PC-8001mkII

USER'S MANUAL

NEC

PC-8001MK2-UM

PTS-120A

御注意

(1)本書の内容の一部又は全部を無断転載することは禁止されています。

(2)本書の内容に関しては将来予告なしに変更することがあります。

(3)本書は内容について万全を期して作成いたしましたが，万一御不審な点や誤り，記載もれなどお気付きのことがありましたら御連絡下さい。

(4)運用した結果の影響については(3)項にかかわらず責任を負いかねますので御了承下さい。

表紙の説明

表紙に使っている写真は，**PC-8001MK**IIのグラフィックス機能を使って描いたものです。

PC-8001mkII

USER'S MANUAL

はじめに

NEC は，マイコンのトレーニングキットとして TK-80 を発売して以来，TK-80BS，TK-85，PC-8000 シリーズ，PC-6000 シリーズそして PC-8800 シリーズと，パーソナルコンピュータ時代の基礎を築いてまいりました。

PC-8001MKII は，ご好評をいただいております PC-8001 をもとに，ご要求の多い機能をもり込んで開発されました。PC-8001 の姉，PC-8801 の妹にあたる機種です。

注　意　事　項

1．使用上のご注意

(1)　ケーブルの抜き差しは，電源を切ってから行なってください。

(2)　スロットバスへのボードの抜き差しは，必ず電源を切ってから行なってください。

(3)　PC-8881(８インチフロッピィディスクユニット)のインタフェースカードは，スロットバス＃2(下側)に実装してください。

(4)　PC-8012-03(音声認識ボード)を使用する場合は,ディップスイッチの２を「on」にしてください。

(5)　PC-8012-02(32Kバイト増設RAMボード)を使用するときはディップスイッチの２を「on」にしてください。

(6)　PC-8001用のソフトウェアを使用するときは，N-BASICモードで動作させてください。モードの切り替えは，PC-8001MKⅡ本体背面のディップスイッチ８で行います。

2．電源に関するご注意

(1)　電源スイッチを一度「off」にしたときは５秒以上経ってから「on」にしてください。電源スイッチが「on」の状態で電源プラグを抜いたときも同様に５秒以上経ってから電源プラグを差し込んでください。

(2)　電源は必ず AC100V (50Hz あるいは 60Hz) を使用してください。

(3)　電源コードを抜き差しする場合は必ずプラグのところを持って行ってください。

3．保管および使用環境に関するご注意

(1)　PC-8001MKⅡは温度上昇を防ぐため，ケースに通風孔が開けてありますので，通風孔をふさいだり，風通しの悪い場所でのご使用をさけてください。また PC-8001MKⅡを極端な高温下や低温下または温度変化の激しい場所で保管及び使用することはさけてください。

(2) PC-8001MK II を直射日光の当る場所や発熱をする器具の近くで保管および使用することはさけてください。

(3) PC-8001MK II を極端に湿気の多い場所や，ほこりの多い場所で保管および使用することはさけてください。

(4) PC-8001MK II は精密な電子部品でできていますので，衝撃を加えたり，衝撃，振動の加わる場所での保管および使用することはさけてください。

(5) PC-8001MK II の内部に水や液状のもの，金属類が入った状態でご使用になりますと危険ですので，異物が入らないようご注意ください。

(6) 薬品の雰囲気中や薬品に触れる場所での保管および使用することは避けてください。

(7) PC-8001MK II を解体した状態で保管および使用することは，故障や感電の原因になりますのでおやめください。

(8) PC-8001MK II の上に重い物を置いた状態で保管および使用することはさけてください。

(9) ラジオやテレビなどのすぐそばで使用しますと，ラジオやテレビに雑音が入ることがあります。また，強い磁界を発生する装置などが近くにありますと，逆に PC-8001MK II に雑音が入ってくることがあります。このような場合は離してご使用ください。

(10) PC-8001MK II は，本体底部にアース端子が備わっています。アース線が配線されている場所ではアースを取ることをお勧めします。

4．その他

(1) PC-8001MK II の汚れはやわらかい布に水または洗剤を含ませて軽くふいて下さい。ベンジン，シンナーなど（揮発性のもの）や薬品を用いてふいたりしますと変形や変色の原因になることがあります。また殺虫剤などをかけた場合でも変形や変色の原因になることがありますので，ご注意ください。

5．異常，故障の場合

⑴　故障や異常（臭いがしたり，過熱していたり）に気付いたときは，直ちに電源コードのプラグを抜いて，お買い求めの販売店あるいはもよりの **Bit-INN** にご相談ください。

PC-8001MKII
ユーザーズマニュアル

目　次

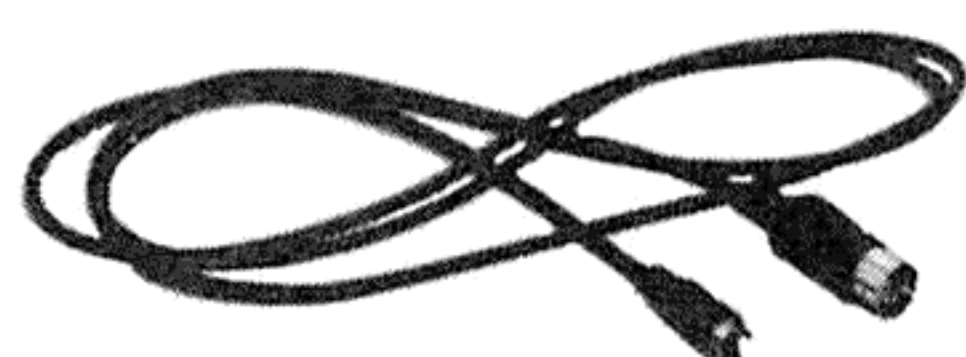
第1章
概　　説

この章では，**PC-8001MKⅡ**（略して **MKⅡ** と呼びます。）の付属品の使い方と，PCシリーズのなかでの**MKⅡ**の位置付けについて説明します。

1.1　付属品の使い方

MKⅡには，図1.1にあげた付属品が添付されています。

PC-8001 MKⅡ

①**PC-8001MKⅡユーザーズマニュアル**
②**N_{80}-BASIC リファレンスマニュアル**
③**N_{80}-BASICリファレンスブック**

④**モノクロディスプレイ用ケーブルPC-8092**
⑥**デモテープ**
⑤**カセットテープレコーダ用ケーブルPC-8093**

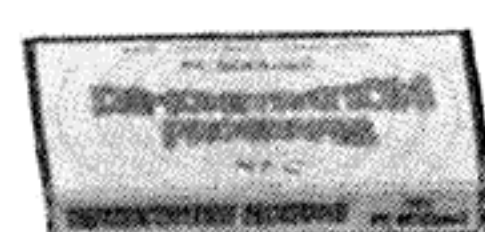
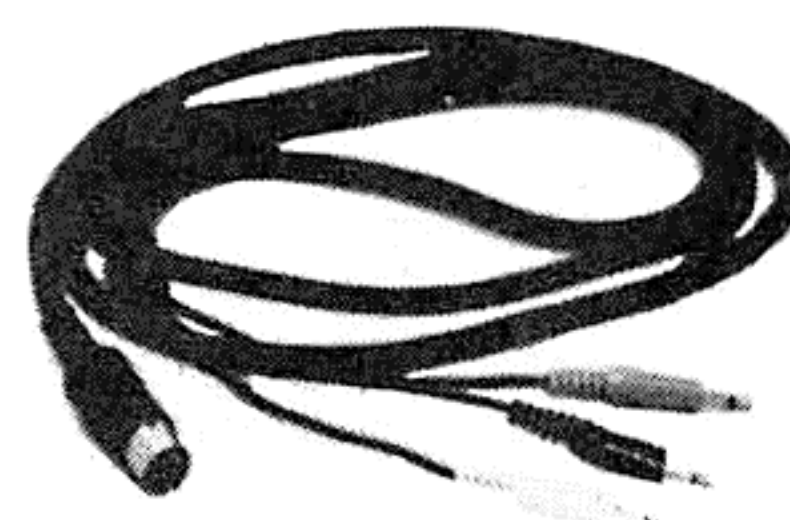



図1.1　**MKⅡ**の付属品

① **PC-8001MKII ユーザーズマニュアル**

このマニュアルは，**MKII** をはじめてお使いになるときに必ず読んでください。

周辺装置の接続のしかた，基本的な操作法，ソフトウエアとハードウエアの概略など，**MK II** を使ううえで，ぜひ知っておいていただきたいことが書かれています。

MKII のプログラミング言語である **N80-BASIC** を使って実際にプログラムを作る時は，「N80-BASIC リファレンスマニュアル」をごらんください。

② **N80-BASIC リファレンスマニュアル**

MKII のプログラミング言語である **N80-BASIC** の文法書です。**N80-BASIC** の命令や関数を，辞書風に説明してあります。

③ **N80-BASIC リファレンスブック**

「N80-BASIC リファレンスマニュアル」をコンパクトにまとめた小冊子です。**MKII** のそばにおいてお使いください。

④ **モノクロディスプレイ用ケーブル**（**PC-8092**）

MKII と専用のモノクロディスプレイを接続するためのケーブルです。

⑤ **カセットテープレコーダ用ケーブル**（**PC-8093**）

MK II とカセットテープレコーダを接続するためのケーブルです。

⑥ **デモテープ**

MKII を紹介するためのデモンストレーションプログラムが書き込まれています。テープに書いてある説明をよく読んでお使いください。

1.2 このマニュアルの内容と使い方

このマニュアルは，梱包箱をあけたあとはじめに読んでいただきたいマ

ニュアルです。第1章から順に読んでいただければ，周辺装置の接続のしかた，起動方法，基本的な操作方法，**MKII** の機能などがご理解いただけると思います。

第5章から10章までは，プログラミングの道具となる**N80-BASIC**の概要，ターミナルモードの使い方などのソフト編です。

第11章からは，ハードウエア編になっています。

システムを組むまで……………………第1章，第2章

起動と基本操作…………………………第3章，第4章

BASICでプログラミングするとき

　グラフィックス機能を使うとき…… 第5章，第6章

　フロッピィディスクユニットを
　お持ちでない方は …………………… 第5章，第7章

　フロッピィディスクユニットを
　お使いになる方は…………………… 第5章，第8章

ターミナルとして使うときは………… 第9章

機械語でプログラミングするときは…. 第5章，第10章，第12章，付録

ハードウェアについて知りたいときは… 第11章，第13章，第14章

漢字を扱うときは ……………………… 第15章

1.3　このマニュアルで使われている記号

このマニュアルでは，わかりやすく説明するために，次の記号を使っています。

RETURN　リターンキーの入力を表わします。

Note　ぜひ覚えておいていただきたい事柄が書かれています。

Sample　使用例などを表わします。

XXXXH　　数字にHをつけて 16 進数を表わします。

　　(Sample) FFH＝255

△　　スペースを表わします。

(サンプルプログラム)　　サンプルプログラムであることを表わします。

(PC-8001とくらべて)　　**MK II** と **PC-8001** をくらべてどこがちがうかを表わします。

(注　意)　　特に注意が必要な事柄であることを表わします。

(参　照)　　参照していただきたい関連項目を表わします。

(書　式)　　BASIC の命令や関数の書式を表わします。

　　(書　式)を表現するときには，次の記号を使っています。

　　＜　＞　　＜　＞で囲まれた項目は，その定義内容を具体的に指定します。その他の項目は，そのまま入力します。

　　〔　〕　　〔　〕で囲まれた項目は，省略できることを意味します。

　　|a b|　　a または b を意味します。

1.4　PCシリーズの中で

この節では，NECのPC-8000シリーズ，PC-8800シリーズ，PC-6000シリーズのパーソナルコンピュータと比較して**MKII**がどこに位置づけられるかを，特にソフトウェア上の互換性を中心に説明します。

PCシリーズのパソコンをまったく使ったことがなく，**MKII**がはじめてという方は，必要に応じて参照してください。

PC-8001

PC-6001

PC-8801

PC-8001MK II

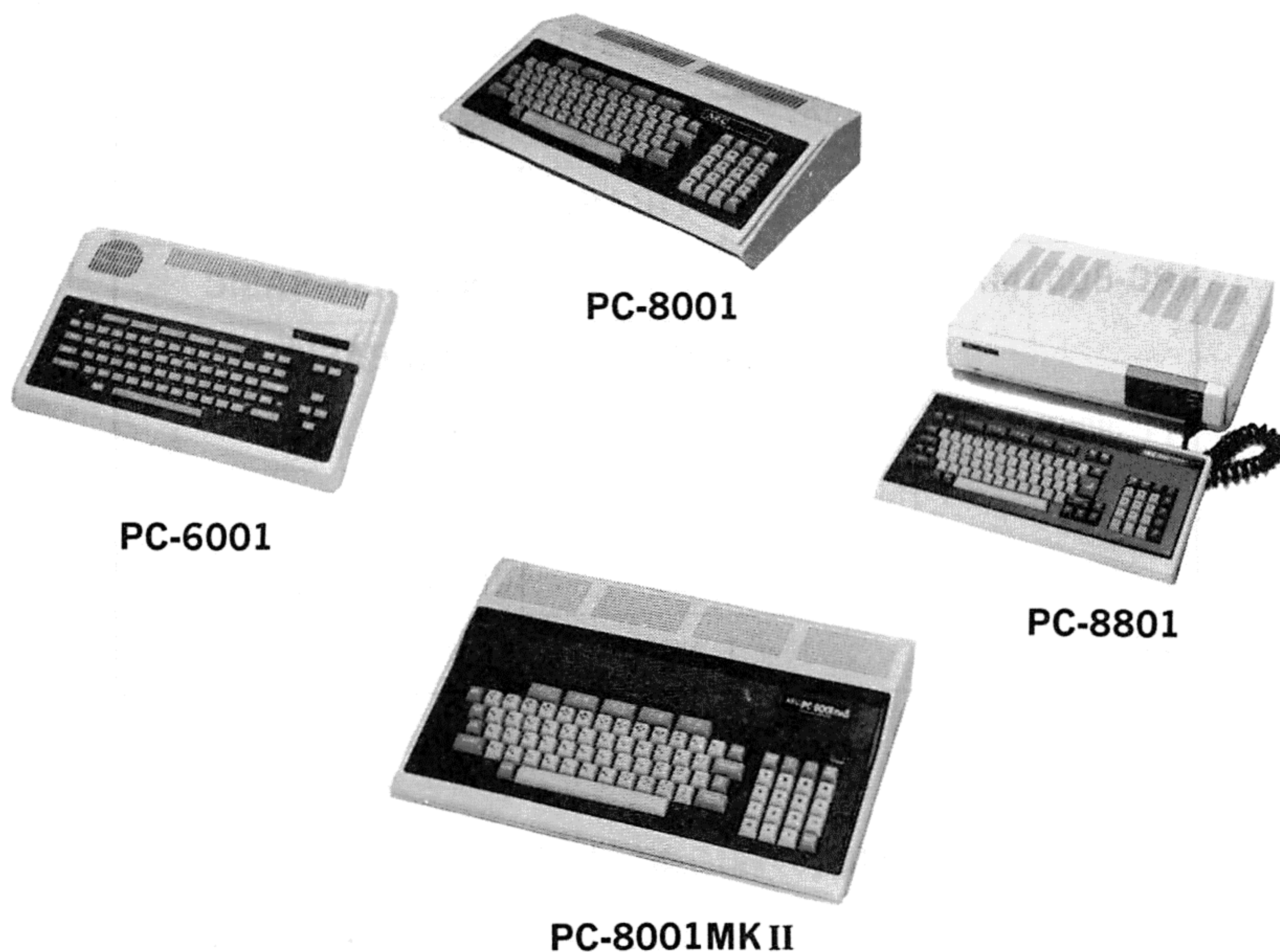
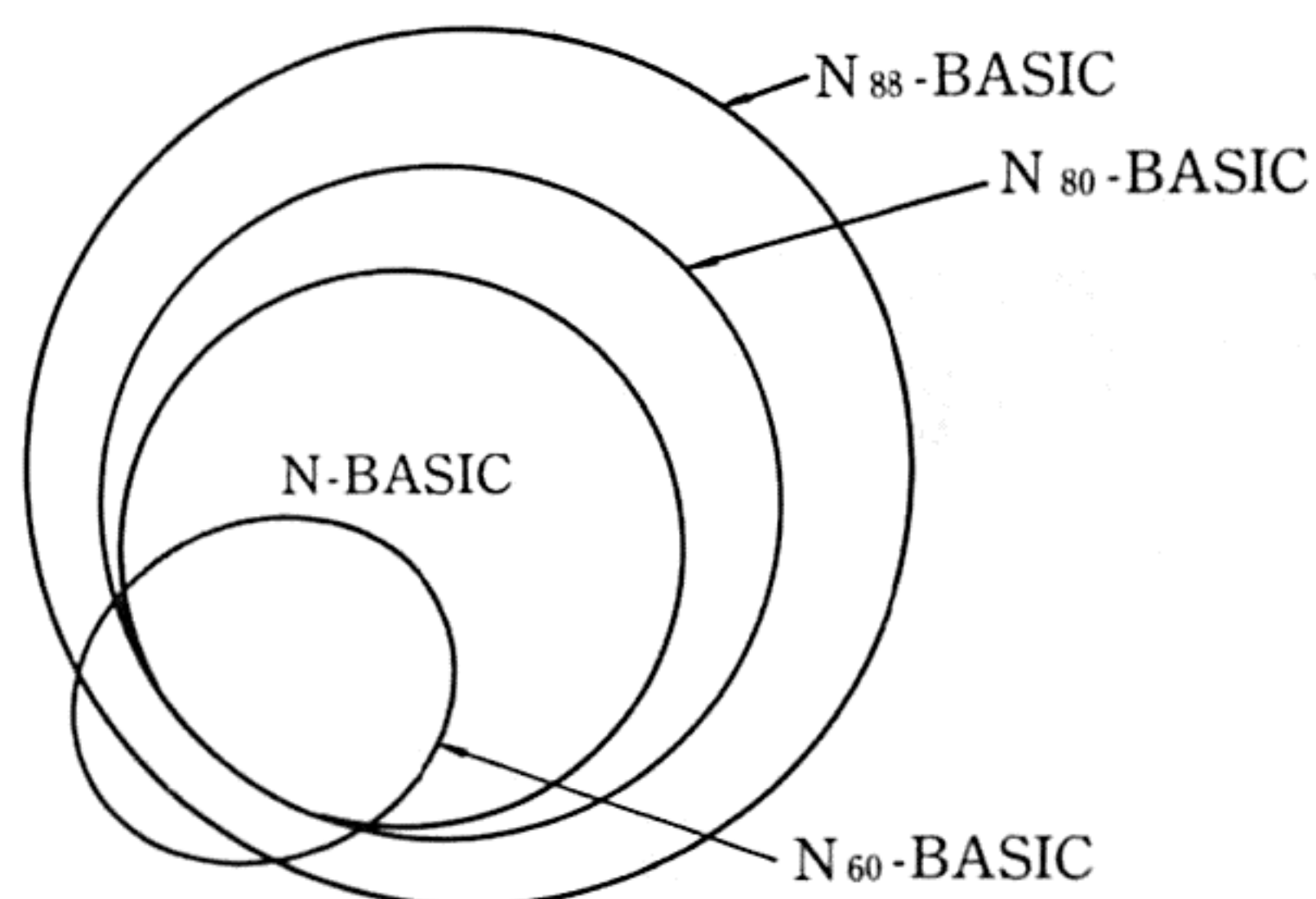



図１．２　PCシリーズの中で

BASIC の文法の互換性

PC シリーズの各製品に搭載されている BASIC はすべて MICROSOFT 社の BASIC をもとにつくられたものです。したがってこれらの BASIC は，文法の核となっている部分は共通で，その核にそれぞれのハードウエアの特長を生かした命令や関数が追加されてできています（図 1.3 参照）。

N_{88}-BASIC

N_{80}-BASIC

N-BASIC

N_{60}-BASIC

図１．３　各BASICの機能の包含関係

表1.1　PCシリーズのBASICファミリー

機種名	BASICの種類	
	ROMモードのBASIC	DISKモードのBASIC
PC-8001MKII	**N_{80}-BASIC**	**N_{80} DISK-BASIC**
	N-BASIC	**DISK-BASIC**
PC-8001	N-BASIC	DISK-BASIC
PC-8801	N_{88}-BASIC	N_{88} DISK-BASIC
	N-BASIC	DISK-BASIC
PC-6001	N_{60}-BASIC	(N_{60}-拡張BASIC)

表 1.1 には PC シリーズの BASIC ファミリーをあげました。この表の ROM モードの BASIC の欄にあげた N_{80}-BASIC などは狭い意味で使っています。広い意味で使うときは，ROM モードと DISK モードをあわせて **N_{80}-BASIC** などと使います。

Note **MKII** の **N_{80}-BASIC** は，**N-BASIC** の機能に，**N_{88}-BASIC** の機能のいくつかを追加して設計されています。追加された機能については第5章を参照してください。

ファイル形式の互換性

プログラムやデータをカセットテープやフロッピィディスクに記録した意味を持つ情報の集まりをファイルといいます。ファイルにはプログラムファイルとデータファイルとがあります。

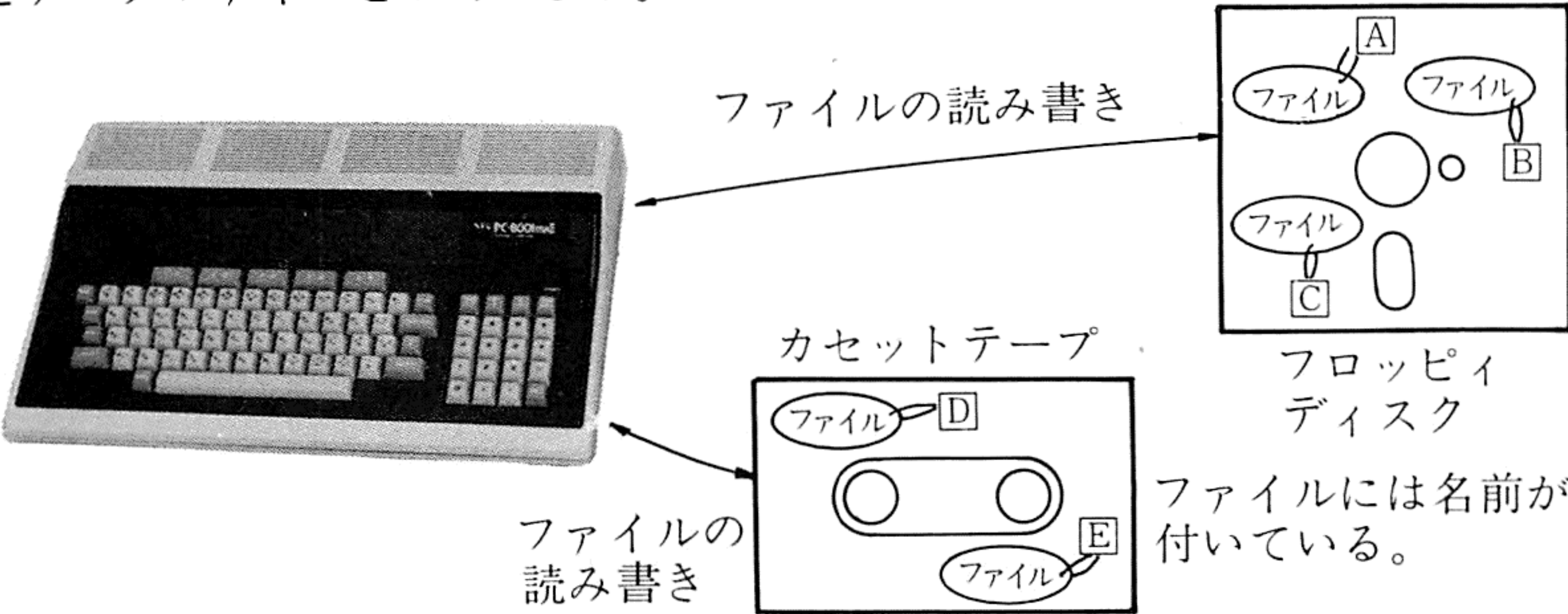



図1.4　ファイル

ファイルには名前がつけられていて（**ファイル名**），**MK II**は，BASIC のもとで，これらのファイルの読み書きをします（カセットテープ上のデータファイルだけは例外でファイル名がありません）。ここでは，これまでの PC シリーズの 各 BASIC の もとで作ったファイルと，**MKII**の **N80-BASIC** のもとで作ったファイルとの互換性について説明します。

1．データファイルの場合

データファイルとは，**PRINT#**，**INPUT#**，**PUT**，**GET**などの命令を使って，カセットテープやフロッピィディスクに読み書きするファイルです。

PC シリーズの各 BASIC は，完全な互換性をもったデータファイルを作ります。したがって，**MKII**と，**PC-8001**，**PC-8801**，**PC-6001** は，互いにデータを交換しあうことができるわけです。

（注　意）ただし **MKII**と **PC-8801** は 8インチ標準フロッピィディスクが使えますが，**PC-8001** と **PC-6001** は，使用できません。また，**MKII** は，カセットテープ上に 1200bps．(bit per second) で書かれたデータファイルをそのまま読むことはできませんから，**PC-6001** や **PC-8801** とデータ交換する場合は，600bps．で書いたものに限ります。
また，**PC-6001** の **N60-BASIC** は，他の機種と桁数など数値表現がちがいます。このため **PC-6001** とデータ交換ができるのは文字列だけです。

2．プログラムファイルの場合

プログラムファイルとは，**LOAD**，**SAVE** などのコマンドで読み書きするプログラムのファイルです。メモリ上のプログラムをカセットテープやフロッピィディスクに書き込むことを“**セーブする**”，メモリ上に読み込むことを“**ロードする**”といいます。

（注　意）プログラムファイルの場合は，ファイルの互換性がある場合でも，各 BASIC の文法がちがうので，そのまま実行

できないことがあるので注意してください。

① **N-BASIC**のもとで作られたプログラムファイル

N$_{80}$-BASICは，**N-BASIC** の機能を完全に含んでいるので，**N$_{80}$-BASIC**のもとで，**N-BASIC**で作ったプログラムファイルをそのまま使うことができます。

しかし逆に，**N$_{80}$-BASIC** は，**N-BASIC** にない機能をもっていますから，**N-BASIC** のもとでは，**N$_{80}$-BASIC** で使ったプログラムのいくつかを使えないことがあります。

② **N$_{88}$-BASIC，N$_{60}$-BASIC** のもとで作られたプログラムファイル

通常，特に指定しないでプログラムをセーブする場合は，プログラムを BASIC インタプリタ（翻訳機）によって中間コードに変換したものを書き込みます。このようなセーブのしかたを **バイナリセーブ** といいます。それぞれの BASICの インタプリタが異なるため，同じプログラムを作っても，できたプログラムファイルは異なったものになってしまいます（図 1.5 参照）。

人間がつくった
プログラム

```
10 input A
20 print A
30 end
```

N-BASIC, N$_{80}$-BASIC
のインタプリタ

N$_{88}$-BASICの
インタプリタ

N$_{60}$-BASICの
インタプリタ

バイナリセーブした
プログラム（中間言語）

```
29 80 0A 00 85 20 41 00
31 80 14 00 91 20 41 00
37 80 1E 00 81 00 00 00

09 00 0A 00 85 20 41 00
11 00 14 00 91 20 41 00
17 00 1E 00 81 00 00 00

09 C4 0A 00 84 20 41 00
11 C4 14 00 95 20 41 00
17 C4 1E 00 80 00 00 00
```

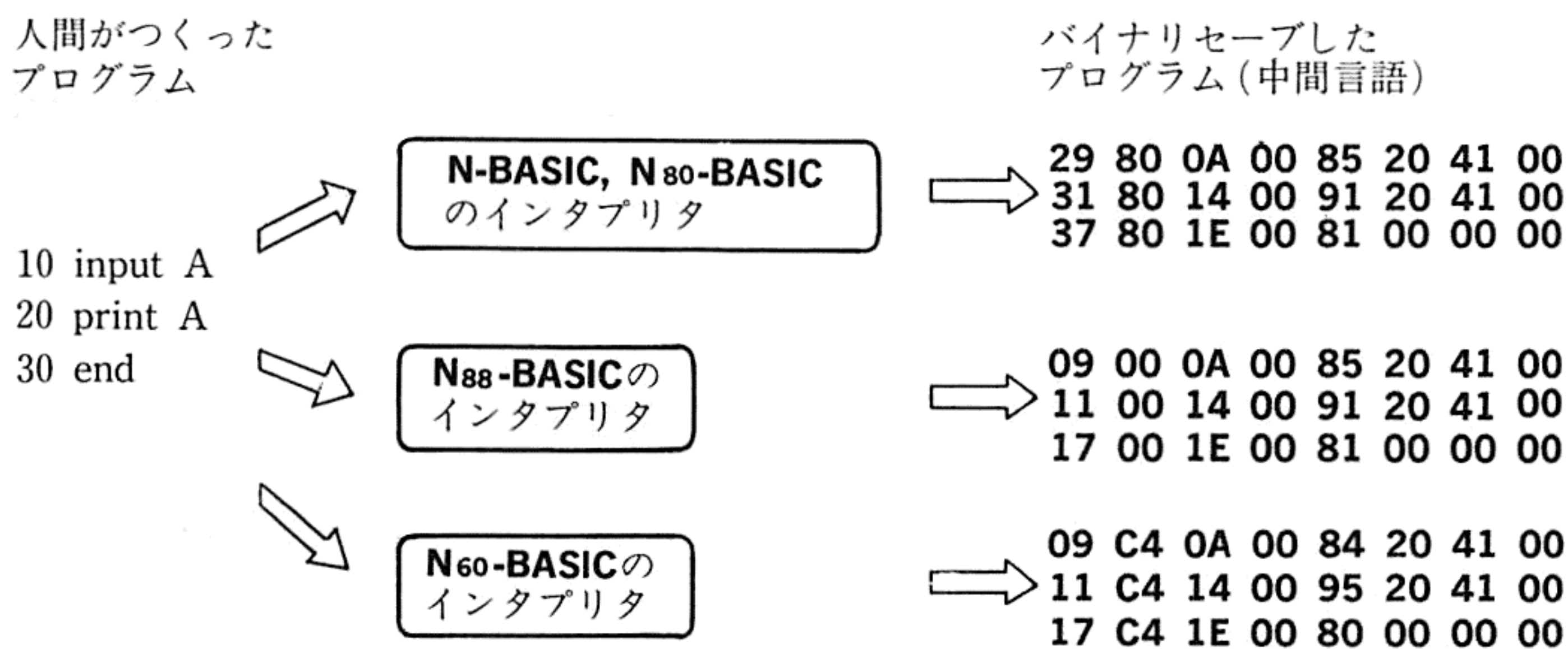



図１．５　中間コードに変換されたプログラムファイル

DISK モードでは，バイナリセーブのほかに，プログラムを構成する１文字１文字をそのままキャラクタコード（アスキーコード）でセーブすることができます。このようなセーブのしかたを**アスキーセーブ**といいます。

アスキーセーブされたプログラムファイルは，図 1.6 のように，

相互にファイルを交換することができます。しかし，機種によって使えない命令もありますので，そのような場合は，変更する必要があります。

N88DISK-BASIC
N60拡張-BASIC
N80DISK-BASIC
セーブする
ロードできる
セーブする
アスキーセーブされたプログラムファイル
アスキーセーブされたプログラムファイル

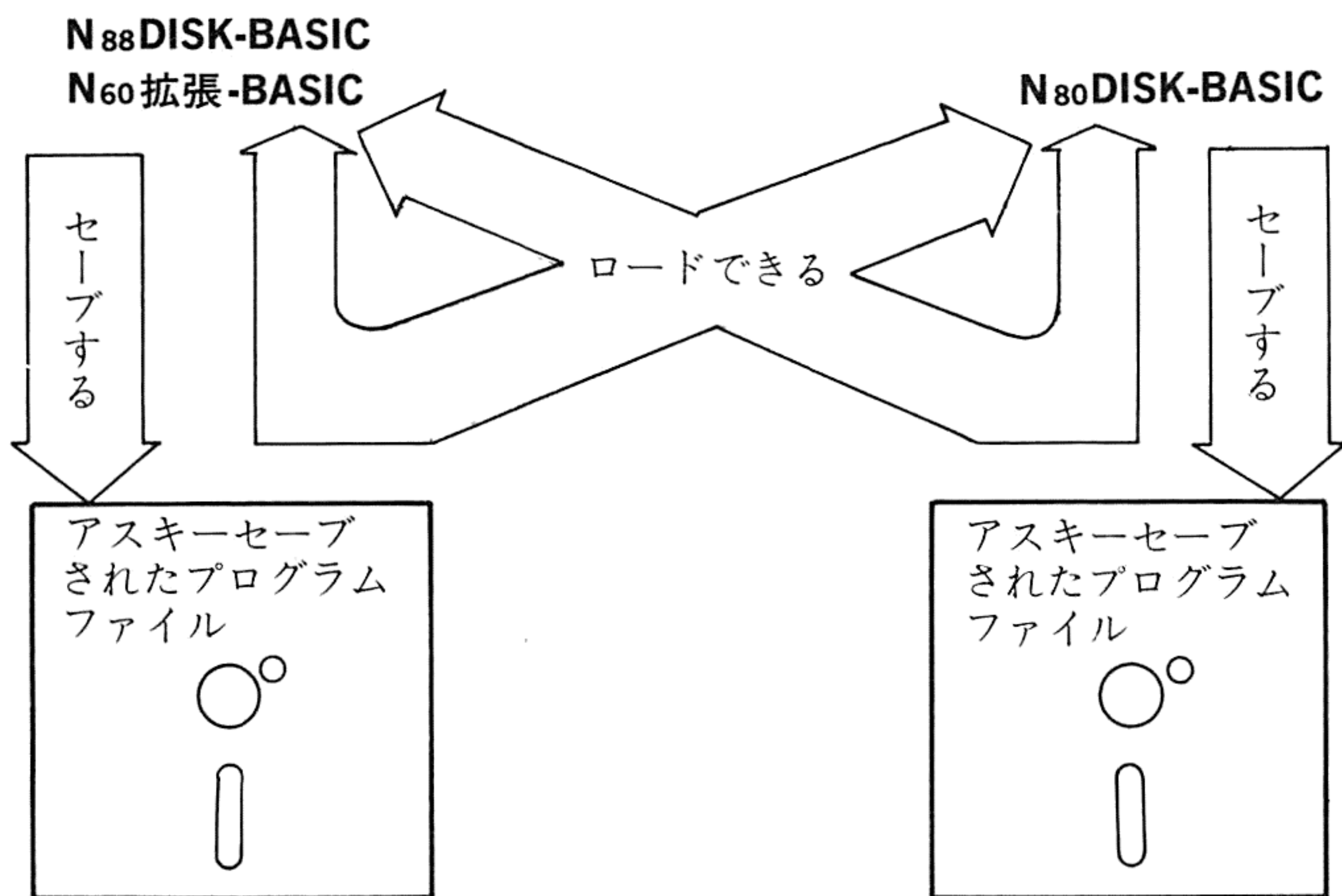



図1.6　アスキーセーブしたファイルの互換性

バイナリセーブされたプログラムファイルを相互に交換した場合は，通常でたらめなプログラムとなってしまいますので，実行できません。

③　機械語のプログラムファイル

MKⅡは，**CMD BLOAD**，**CMD BSAVE**というコマンドを使って機械語のプログラムファイルを扱うことができます。**MKⅡ**と**PC-8801**の機械語プログラム形式は全く同じです。

第2章
システム構成

MKII は、本体だけでは何の役にもたちません。ステレオでもアンプにプレーヤやスピーカをつながなければならないように，**MKII** にも周辺装置を接続してはじめて，使えるようになるのです。このように，必要な周辺装置を接続して，できるセットをシステムといいます。

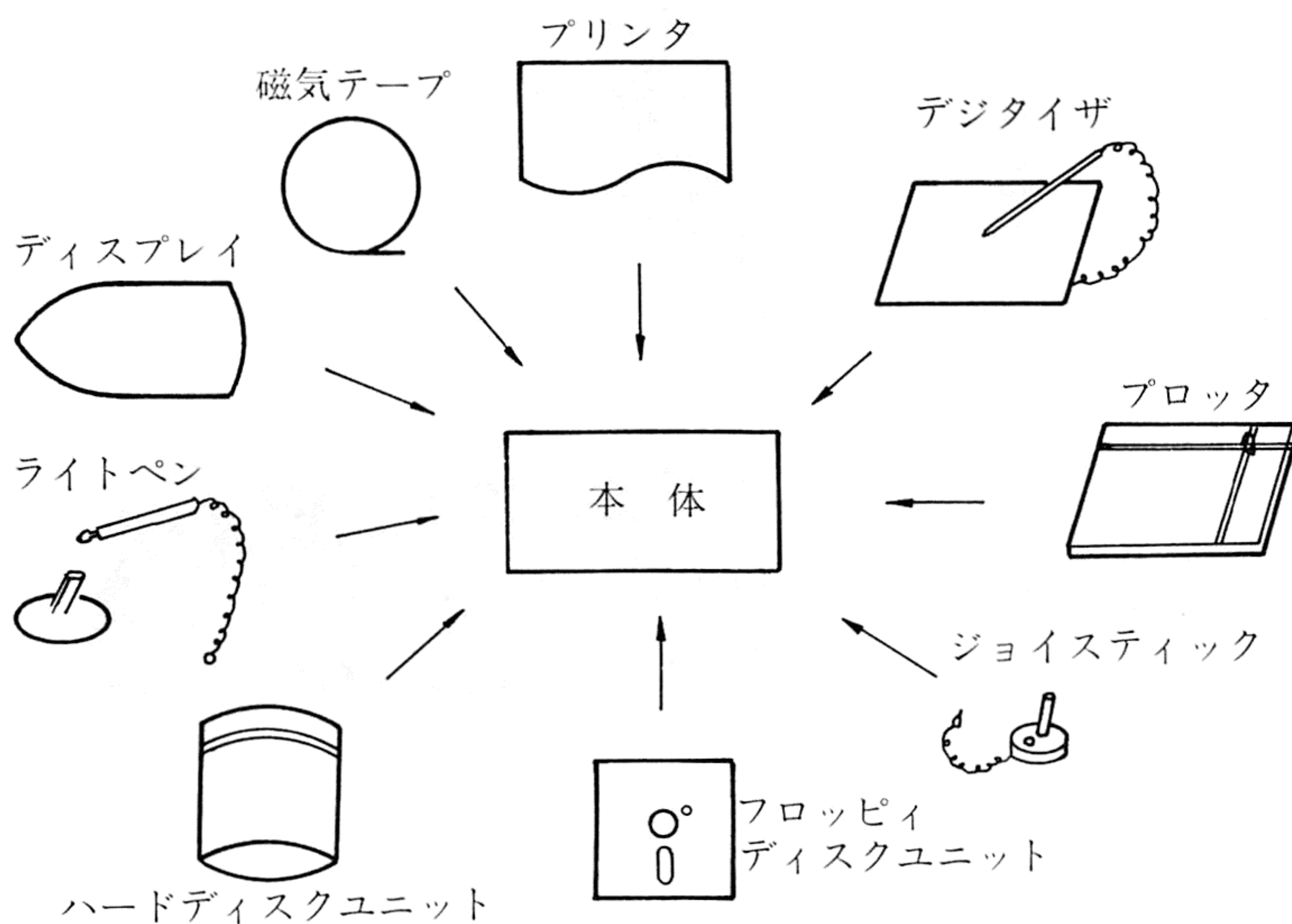



図２．１　本体と周辺装置

図２．１に一般的な本体と周辺装置の関係を示しました。本体にいろいろな周辺装置が接続されています。本体（**CPU** ともよばれます）にあたるのが **MKII** です。

この章では，前半で**MKII**のシステム構成の例を３つあげ，後半で **PC シリーズ**の周辺装置のおもなものを，**MKII**に接続できるかできないかでわけて紹介します。

2.1 システム構成例

MKII は，**PCシリーズ** の豊富な周辺装置を組み合わせることによって，用途に応じたシステムを構成することができます。ここでは3種類のシステム構成例を紹介します。システム構成についての詳しいことは，**Bit-INN** やお買い上げの販売店でご相談ください。

1．最小システム

家庭用テレビ
PC-8044K
付属のPC-8093
PC-6081
PC-8001MK II

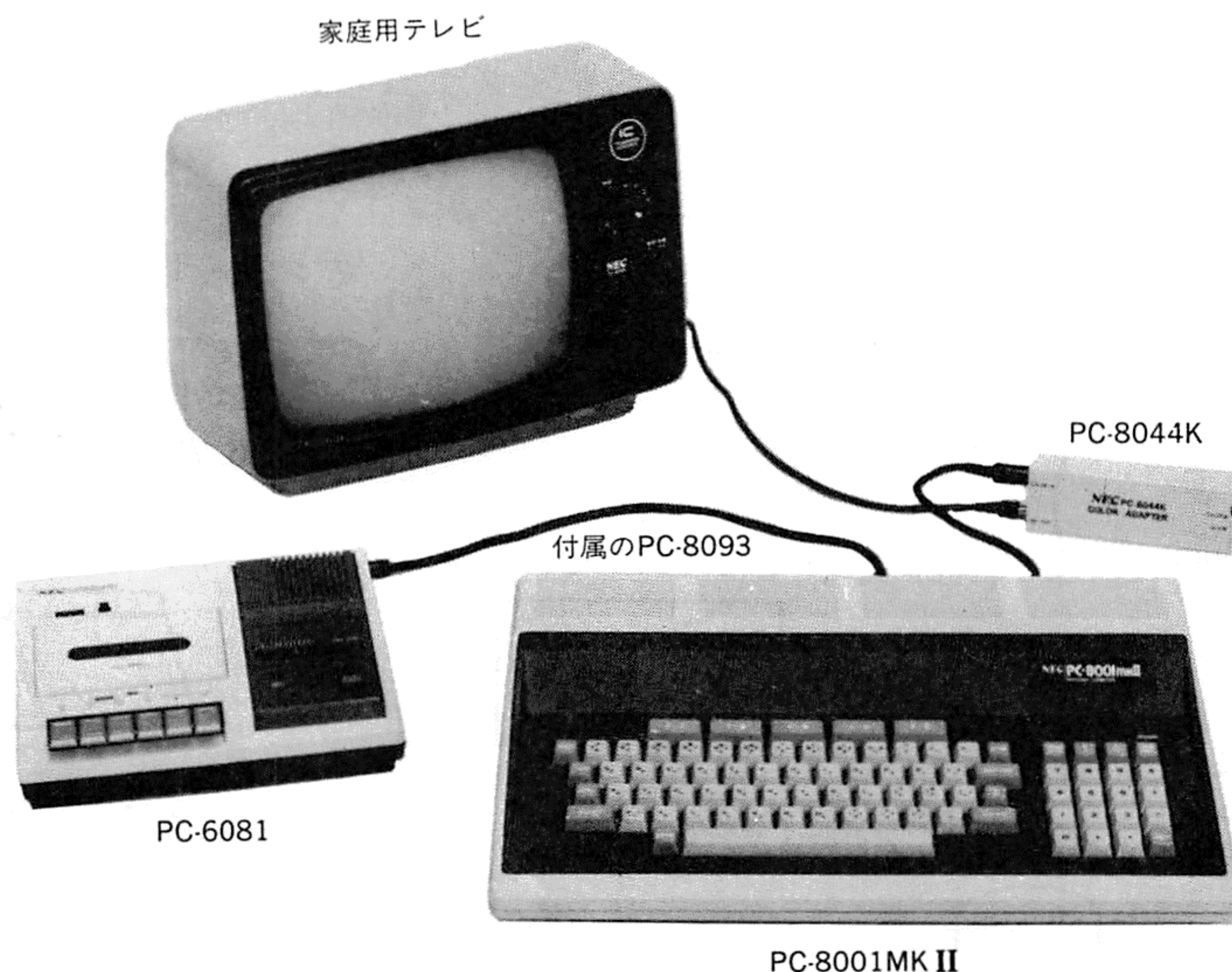



図2．2　最小システム

MKII に，家庭用のテレビ（NTSC* 方式）とカセットテープレコーダ（**PC-6081**）だけを接続した，家庭用のシステムです。家庭用のテレビは，RFモジュレータ（**PC-8044K**）を経由して接続します。

（**注　意**）家庭用のテレビは，解像度が低いので，細かい文字や漢字を表示した場合，見にくいことがあります。そのような場合は，2．2節にあげる専用のディスプレイをお勧めします。

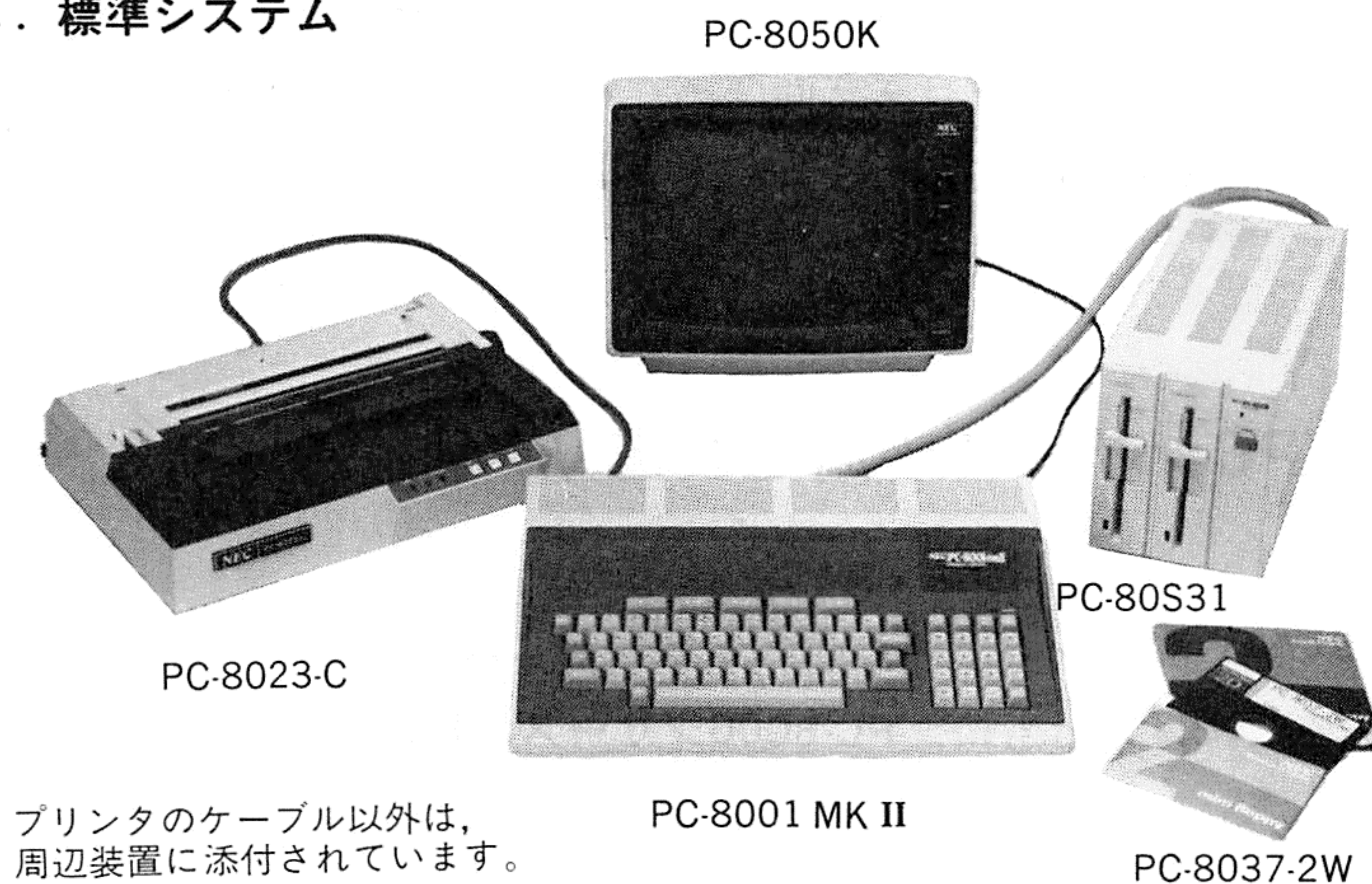
* NTSC (National Television Standard Commitee)
日本のテレビ放送は NTSC 方式を採用しています。

2．標準システム

PC-8050K

PC-8023-C

PC-80S31

PC-8001 MK II

PC-8037-2W

プリンタのケーブル以外は，
周辺装置に添付されています。

図2．3　標準システム

ミニフロッピィディスクユニット(**PC-80S31**)とプリンタ(**PC-8023-C**)を接続し，ディスプレイには**PC-8050K**を使ったシステム構成です。フロッピィディスクユニットの接続により**MK II**の機能は，飛躍的にアップします。

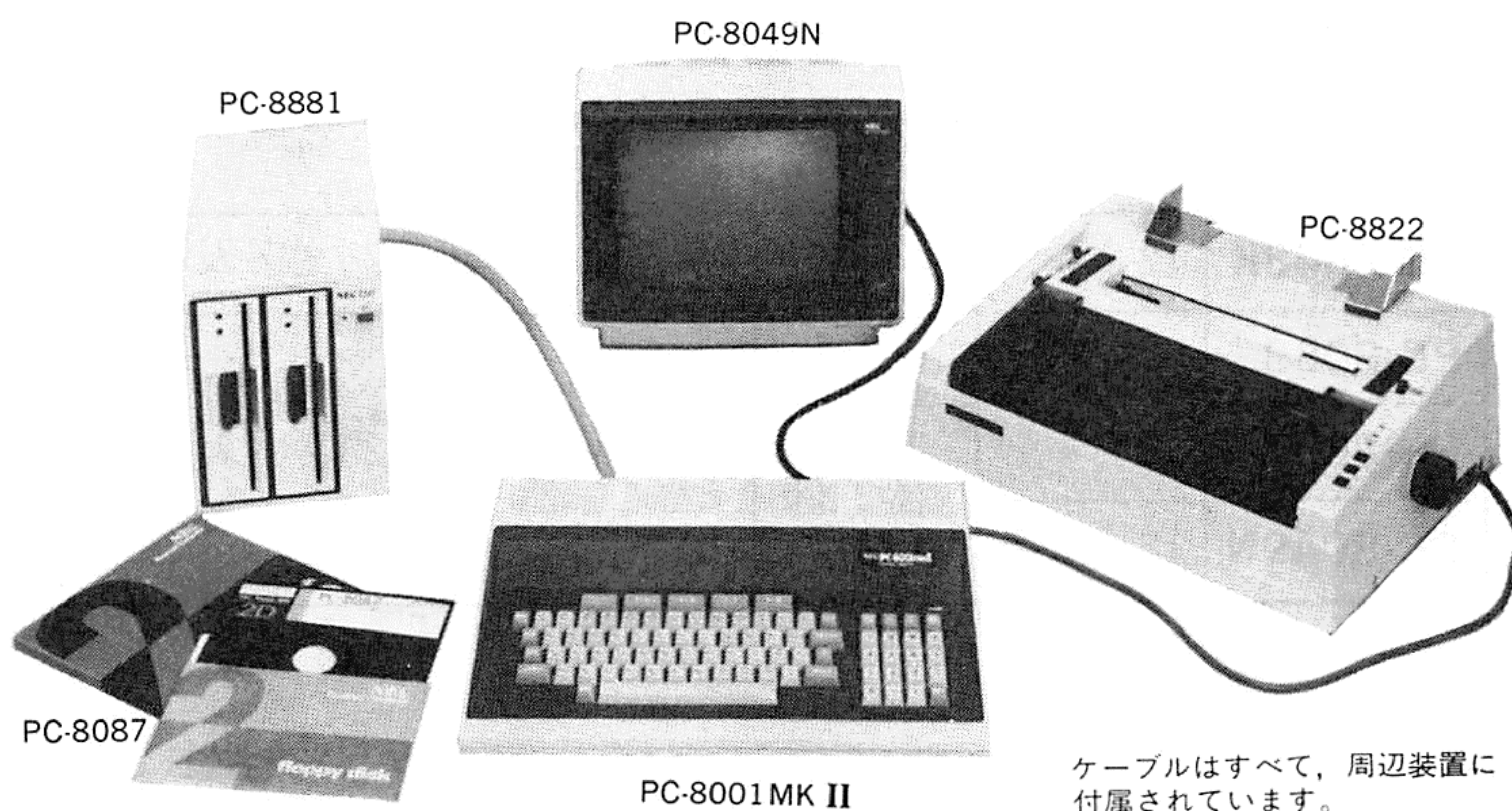
N80DISK-BASICを使うためには，システムディスク(**PC-8037-2W**)が必要です。また**N80DISK-BASIC**のほかにも各種の**DOS**(Disk Operating System)を使うことができます。

3．ビジネス向けのシステム

PC-8049N

PC-8881

PC-8822

PC-8087

PC-8001MK II

ケーブルはすべて，周辺装置に
付属されています。

図2．4　ビジネス向けのシステム

8インチフロッピィディスクユニット(**PC-8881**)と漢字プリンタ(**PC-8822**)を接続したビジネス向けのシステムです。**MKII**の中に漢字ボード(**PC-8001MKII-01**)を組み込めば漢字処理も行えます。

2.2　PCシリーズの周辺装置リスト

MKII は，**PC シリーズ**の周辺装置のほとんどを接続することができます。ここには，おもに，**PC-8000シリーズ**，**PC-8800シリーズ** の製品のなかから1983年2月現在発売されている周辺装置をあげます。

2.2.1　MKIIに接続して使用できるもの

ディスプレイ

R.G.B.セパレート方式のカラーディスプレイと，コンポジットビデオ方式のモノクロディスプレイを接続することができます。水平走査周波数は15KHzです。

カラーディスプレイ

カラーディスプレイには，解像度の異る2種類があります。高解像度のディスプレイは，80×25 文字や漢字の表示ができますが，中解像度のもの(**PC-8048K**)は，見にくいので注意してください。

品　　名	大きさ	ケーブル	備　　考
PC-8048K	12インチ	別売り	中解像度。40×25文字表示可能。
PC-8053	14インチ	別売り	高解像度。80×25文字表示可能。
PC-8052	14インチ	添　付	高解像度。80×25文字表示可能。
PC-8049N	12インチ	添　付	高解像度。80×25文字表示可能。

モノクロディスプレイ

PCシリーズの モノクロディスプレイは，すべて高解像度です。ケーブルは，**MKII** に添付されています。

品　　名	大きさ	色	ケーブル
PC-8046	9インチ	グリーン	**MKII**に添付。
PC-8047	12インチ	アンバー	
PC-8050K	12インチ	グリーン	

家庭用テレビアダプタ PC-8044K

PC-8044K を使って家庭用のテレビを **MKII** のディスプレイとして使うことができます。解像度が低いので，細かい文字の表示には向きません。

ライトペン PC-8045K

PC-8045K は，カラーディスプレイまたはモノクロディスプレイと，組み合わせて使います。ライトペンは，ディスプレイ画面のライトペンでさした位置をキャラクタ単位で読みとるために使います。

カセットテープレコーダ

プログラムやデータのロードやセーブは，家庭用のテープレコーダで充分行えますが，**MKII** 用に新しく購入される場合は，**PC-6000 シリーズ**のデータレコーダをお勧めします。一般のテープレコーダでは，イヤホーン端子にジャックを差し込んでしまうとスピーカから音がでませんが，**PC-6081, PC-6082** ならデータの入っている部分を聞きわけられます。

品　　名	ケーブル	備　　　考
PC-6082	**MKII**に添付。	自動プログラムサーチ機能つきパソコン用テープレコーダ。
PC-6081		パソコン用テープレコーダ。

ミニフロッピィディスクユニット

両面，片面のどちらのタイプも接続できます。**PC-8000 シリーズ**のミニフロッピィディスクユニットは，インテリジェントタイプになっています。

品　　名	ケーブル	システムディスク	備　　考
PC-80S31	添　付	PC-8037-2W（別売）が必要	
PC-8031-2W	別売りのPC-8898が必要		両面倍密度ミニディスクドライブ2個実装。
PC-8031-1W		PC-8037-1W（別売）が必要	片面倍密度ミニディスクドライブ2個実装。
PC-80S32	添　付		PC-80S31の拡張用ドライブ。
PC-8032-2W			PC-8031-2Wの拡張用ドライブ。
PC-8032-1W			PC-8031-1Wの拡張用ドライブ。

標準フロッピィディスクユニット

PC-8800シリーズの標準フロッピィディスクユニットを接続することができます。この場合**MKII**のスロットにインタフェースボードをさして使用します。

品　　名	ケーブル	システムディスク	備　　考
PC-8881	添付	PC-8087（別売）が必要。	2台の8インチ薄形ドライブを実装した標準フロッピィディスクユニットです。（**MKII**とのインターフェースボード付）
PC-8881(1)			PC-8881のドライブを上下左右を逆にしたものです。
PC-8882			PC-8881用の拡張用ユニットです。PC-8881とPC-8882を接続することにより，4Mバイトの記憶容量が得られます。
PC-8882(1)			PC-8881(1)用の拡張用ユニットです。

プリンタ

セントロニクス準拠の インタフェースを採用しているプリンタを接続することができます。

（注　意）**MKII**の**CMD COPY**コマンドで，ディスプレイの画面のコピーがとれるのは，**PC-8023-C**, **PC-8022**, **PC-8021**だけです。ただし，グラフィックスをコピーする場合に，縦横の比がディスプレイ画面

のとおりにならないことがあります。

品　　名	ケーブル	備　　考
PC-8023-C	**PC-8894** (別売り)	9ピンドットマトリクスプリンタ
PC-8822		18ピンドットマトリクス漢字プリンタ
PC-8821	添　付	**PC-8822**から漢字ROMボードをとったもの
PC-8826		カラープロッタプリンタ

プリンタ用別売り品

品　　名	備　　考	
PC-8023-01	**PC-8023C**用インクリボンカートリッジ	
PC-8821-01	**PC-8821**，**PC-8822**用黒色インクリボンカートリッジ	
PC-8821-02	**PC-8821**用漢字ROMボード	
PC-8821-03	**PC-8821**，**PC-8822**用トラクタフィーダアダプタ	
PC-8821-04	**PC-8821**，**PC-8822**用赤/黒インクリボンカートリッジ	
PC-8826-XB	**PC-8826**用ボールペン	X=0：黒色　X=2：赤色
PC-8826-XF	**PC-8826**用サインペン	X=1：青色　X=4：緑色
PC-8826-P3	**PC-8826**用ロール紙	

拡張用ボード

PC-8012に組み込むために作られたボードですが，**MKII**のスロットにさして使うことができます。

品　　名	備　　考
PC-8012-01	ユニバーサルボード
PC-8012-02	32Kバイト増設RAMボード
PC-8012-04	音声録音再生ボード
PC-8012-05	音声認識ボード
PC-8801-02	128Kバイト増設RAMボード

2.2.2 MKIIに接続できないもの，必要ないもの

PC-8851	ディスプレイ	水平走査周波数が異なるので使用できません。
PC-8853	ディスプレイ	
PC-8801-01	PC-8801用漢字ROMボード	MKIIとはサイズが異なるので使用できません。PC-8001MKII-01をご使用ください。
PC-8006	増設RAM	MKIIに標準装備されているので必要ありません。
PC-8033	PC-8001－PC-8031インタフェース	
PC-8012	PC-8001用I/Oユニット	MKIIはこれらの機能のいくつかを標準装備しています。
PC-8013		
PC-8011	PC-8001用拡張ユニット	
PC-8094	プリンタケーブル	コネクタがちがうので使用できません。PC-8894をお使いください。
PC-8095	RS232Cケーブル	MKII用のRS232CケーブルはPC-8895です。
PC-8062	RS232Cケーブル	
PC-8862-01	PC-8062用ROM	ソケットがちがうので接続できません。
PC-8096	GPIBケーブル	MKIIにはGPIB用のインタフェースがついていないので使えません。
PC-8097	GPIBインタフェースセット	

2.2.3 MKIIに接続できるが注意が必要なもの

PC-8031-1V	5インチシングルドライブのディスクユニット	N$_{80}$DISK-BASICには，シングルドライブ用のシステムディスクが用意されていないので使えません。DISK-BASIC(PC-8001のモードで使う)には使えます。

第３章
基本的な操作方法

この章では，①システムを配置する，②周辺装置を接続する，③電源を入れる，④電源を切る，など **MKII** の最も基本的な操作方法について説明します。

3.1　システムの配置

システムの配置場所は，もくじの前にある「**保管および使用環境に関するご注意**」をよく読んで決めてください。

3.2　周辺装置との接続

図３．１のフローチャートと，このマニュアルの最後にある「**周辺装置の接続説明図**」を参考にして，周辺装置を接続してください。

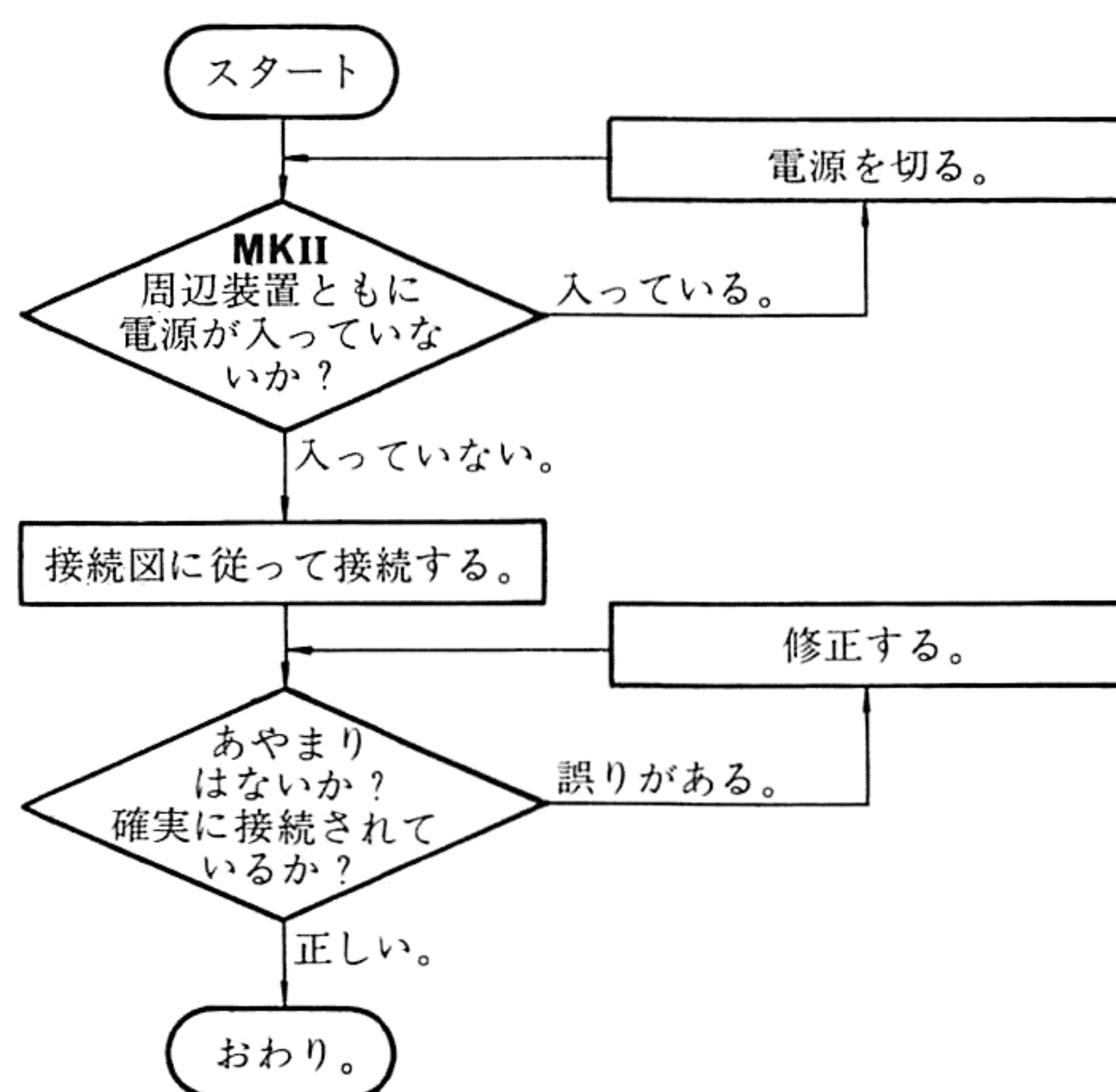



図３．１　周辺装置の接続のフローチャート

(注　意) ①　ケーブルの抜き差しは，必ず電源を off にしてから行う。

②　ケーブルは，逆向きにさせるものがあるので正しくさしたかどうかを必ず確認する。

③　ケーブルは，深く確実に差し込む。

(参　照) 使用するケーブルについては，**第2章システム構成**をごらんください。

3.3　電源の投入と起動テスト

ここでは，フロッピィディスクユニットを接続しないで起動テストを行います。図3.2のフローチャートと，以下の説明に従って行います。

スタート
電源OFFの状態で本体のディップスイッチをセットする。
周辺装置の電源を入れる。
同時でもよい。
システムディスク をセットする。
MK II の電源を入れる。
MK IIのリセットボタンを押す。
正しい動作をしたか？
しない
あやまりはないか？
ある
修正する。
した。
おわり。
ない。
わからない。
販売店に相談する。
[- - -] は，フロッピィディスクユニットを接続してある場合のみ。

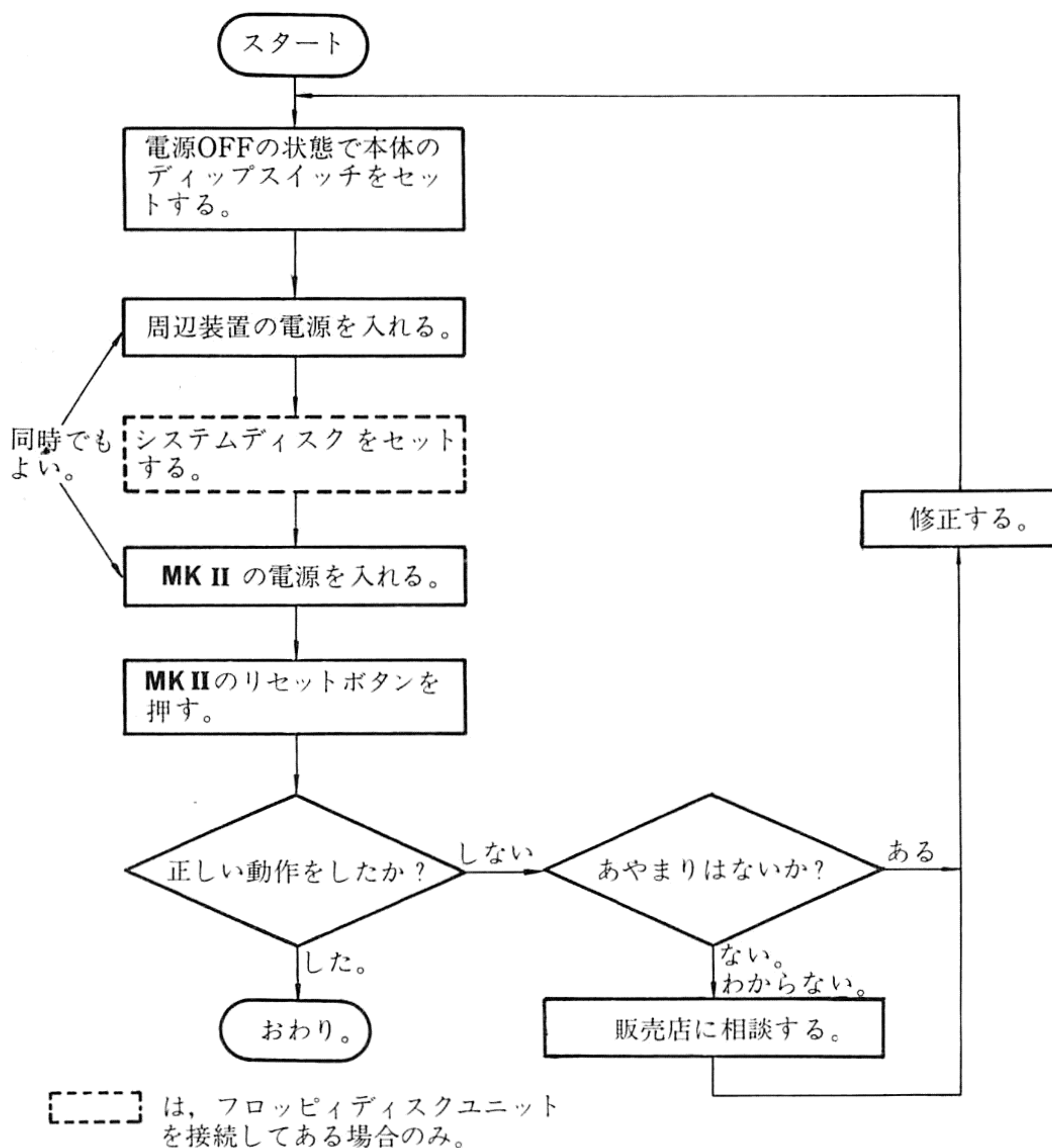



図3.2　起動テストのフローチャート

1．電源を入れる手順

① フロッピィディスクユニット以外の周辺装置を接続します。

② 電源を off にした状態で，**MK II** の背面のディップスイッチ 8 を off にセットします。ディップスイッチ 8 を off にすると **N80-BASIC**, on にすると **N-BASIC** が選ばれます。

ディップスイッチ 1 から 7 までは，on, off どちらになっていてもかまいません。

(参　照) ディップスイッチについては，**付録 3　ディップスイッチとジャンパスイッチの使い方**をごらんください。

(注　意) ディップスイッチとジャンパスイッチの切り換えは，電源を off にした状態で行ってください。

③ 周辺装置の電源を on にします。

④ **MK II** の電源を on にします。

⑤ リセットボタンを押します。これで電源投入は終わりです。

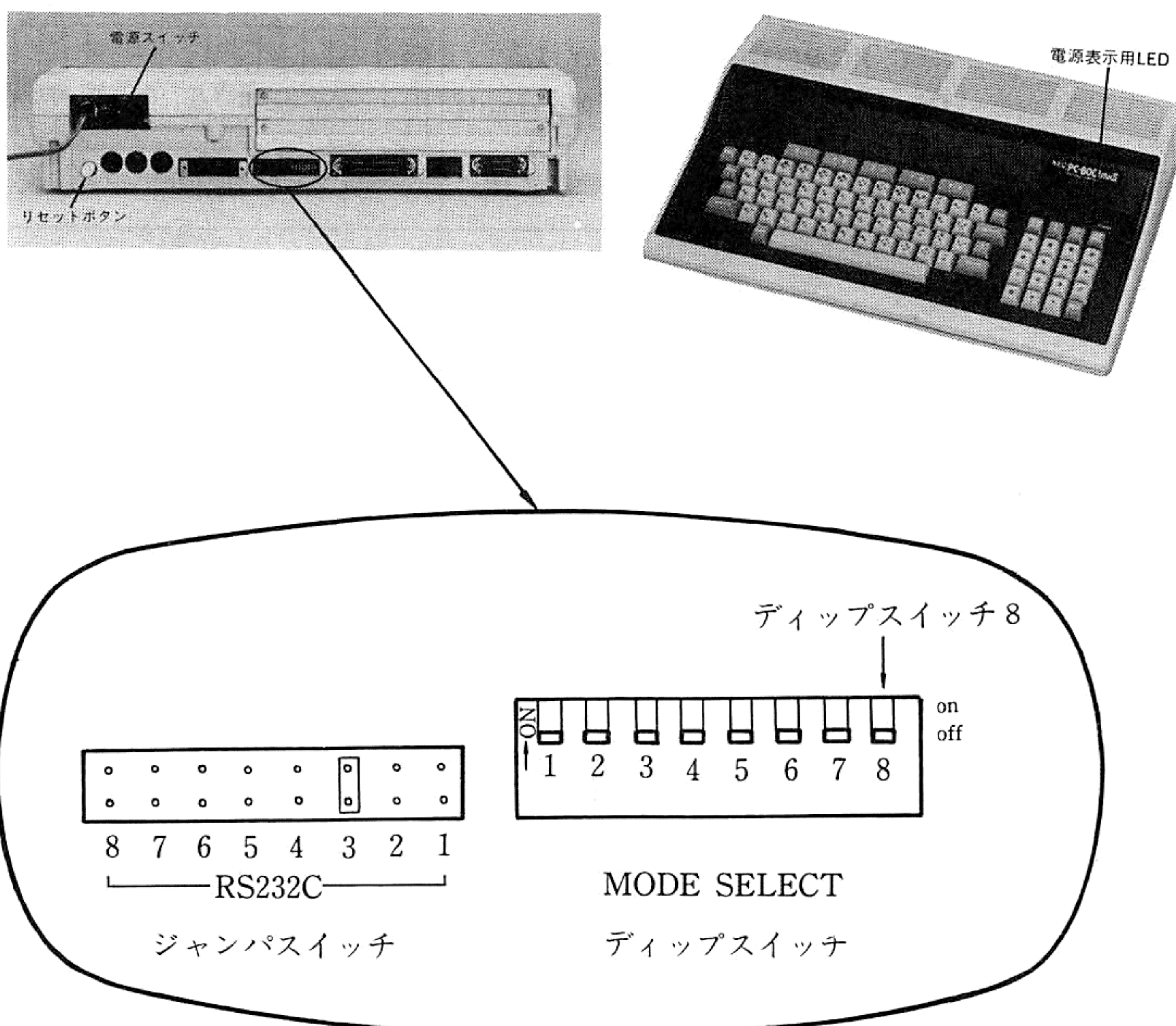



図 3．3　スイッチ類の位置

2. 電源投入時の画面表示

電源を投入すると（操作中にリセットボタンを押したときも）ディスプレイには次の2種類のうちいずれかのメッセージが表示されます。（バージョンによって多少異なるメッセージが表示されることがあります。）これが「起動した」状態です。メッセージが表示されたら起動テストは合格です。

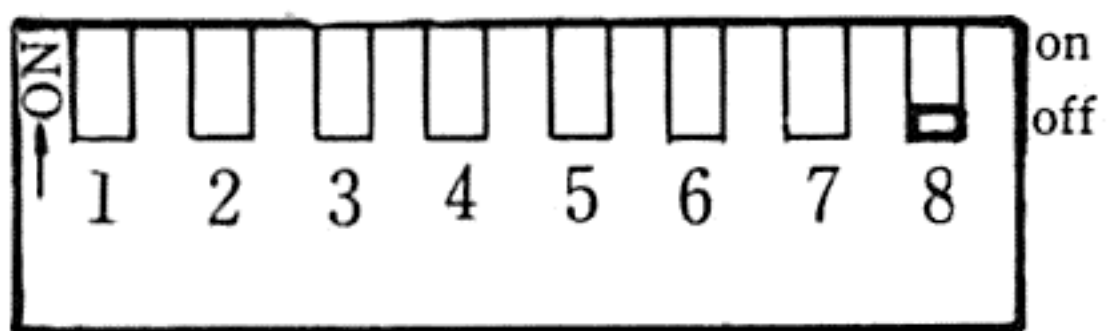
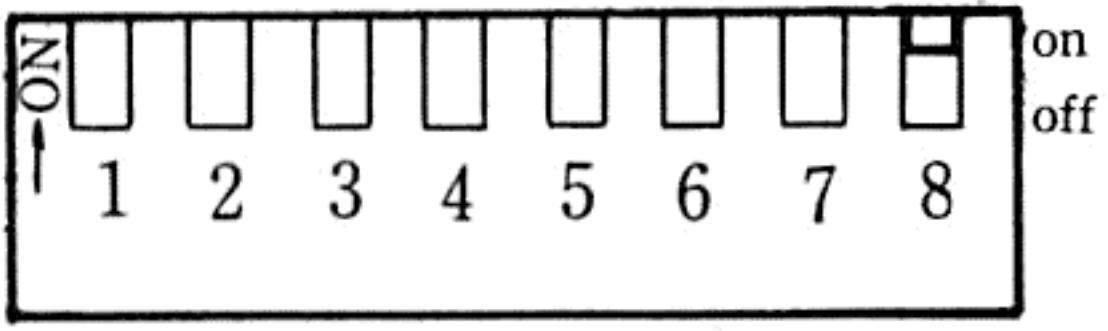
MK II は，2つの BASIC モードをもっています。電源を投入したとき（またはリセットボタンが押されたとき）は，ディップスイッチ8によって，どちらかの BASIC が起動されます。

BASICのモード	メッセージ	ディップスイッチ
N_{80}-BASIC	NEC N80 BASIC Ver 1.0 Copyright 1979(C) by Microsoft Enhanced 1982 by NEC Ok □	ON ↑ 1 2 3 4 5 6 7 8 on off スイッチ8をoffにしたとき
N-BASIC	NEC PC-8001 BASIC Ver 1.3 Copyright 1979(C) by Microsoft Ok □	ON ↑ 1 2 3 4 5 6 7 8 on off スイッチ8をonにしたとき

図3.4　ディップスイッチ8とBASICモード

メッセージの下に表示されている“**Ok**”は，“さあ 準備完了，BASIC の命令を入力してください。”という印です。キーボードのキーを押すとその文字がカーソルの点滅している位置に表示されていくはずです。

3. 正常に起動しない場合の対処方法

① 画面の同期があわず，流れてしまうとき

ディスプレイの電源の on，off を繰り返して行ってみてください。それでも同期がとれない場合は，ディスプレイの水平同期ツマミを回して調整してください。

② 画面に何も表示されない場合

ディスプレイの明るさ（BRIGHT）調整ツマミを回してみてくだ

さい。それでも何も表示されない場合は，全装置の電源を off にして，もう一度正しく接続されているか調べてください。

以上のチェックを行っても画面にメッセージが表示されず，その原因がわからない場合は，お買いあげの販売店か，もよりの **Bit-INN** にご相談ください。

3.4　カレンダ時計

MKII には，バックアップ電池つきのカレンダ時計が内蔵されています。

1．カレンダ時計のあわせかた

“**Ok**”の 表示の あとに次のようにキーボードから入力（以下キー入力と呼びます。）することによってカレンダ時計のセットができます。

(**Sample**) 1983 年 1 月 1 日午後 1 時 15 分にセットする場合

```
date$="83/01/01"  [RETURN]
time$="13:15:00"  [RETURN]
```

(**注　　意**) **$**, **"**, **/** などのキー 入力方法がわからない方は**第4章キーボードの使い方とスクリーンエディタ**を先にお読みください。

カレンダ時計に，一度値をセットすると，**MK II** の電源を off にしても，バックアップ電池の機能により時間が刻まれていきます。

2．カレンダ時計の値の表示のしかた

ディスプレイにカレンダ時計の値を表示するには，次のようにキー入力します。

```
? date$ [RETURN]
? time$ [RETURN]
```

3．バックアップ電池

カレンダ時計のバックアップ電池は，**MKII** の電源を連続して 40 時間以上 on にしておくと完全に充電されます。完全に充電されたあとさらに **MKII** の電源を on にしていても，バックアップ電池の機能を損うことはありません。完全に充電されたバックアップ電池は，**MK II** の電源を off にしたままでも，約 2 か月間カレンダ時計を動かし続けます。バックアップ電池を完全に充電しなくても，1 日に 30 分間程度 **MKII** の電源を on にすれば，カレンダ時計は動き続けます。

3.5　スピーカ

MK II にはスピーカが内蔵されています。次のようにキー入力すれば，約 0.5 秒間ブザーが鳴ります。

beep **RETURN**

3.6　メモリテスト

MK II は，自分自身のメモリのテストをすることができます。メモリテストの方法については，**第 10 章　モニタ**　でくわしく説明します。

3.7　STOP キーとリセットボタンの使い方

この節では，**MK II** の操作中に，キーボードからの入力ができなくなってしまったときどうしたらよいかを説明します。　このような場合は，**STOP** キーとリセットボタンを使います。フローチャートをごらんくだ

さい。

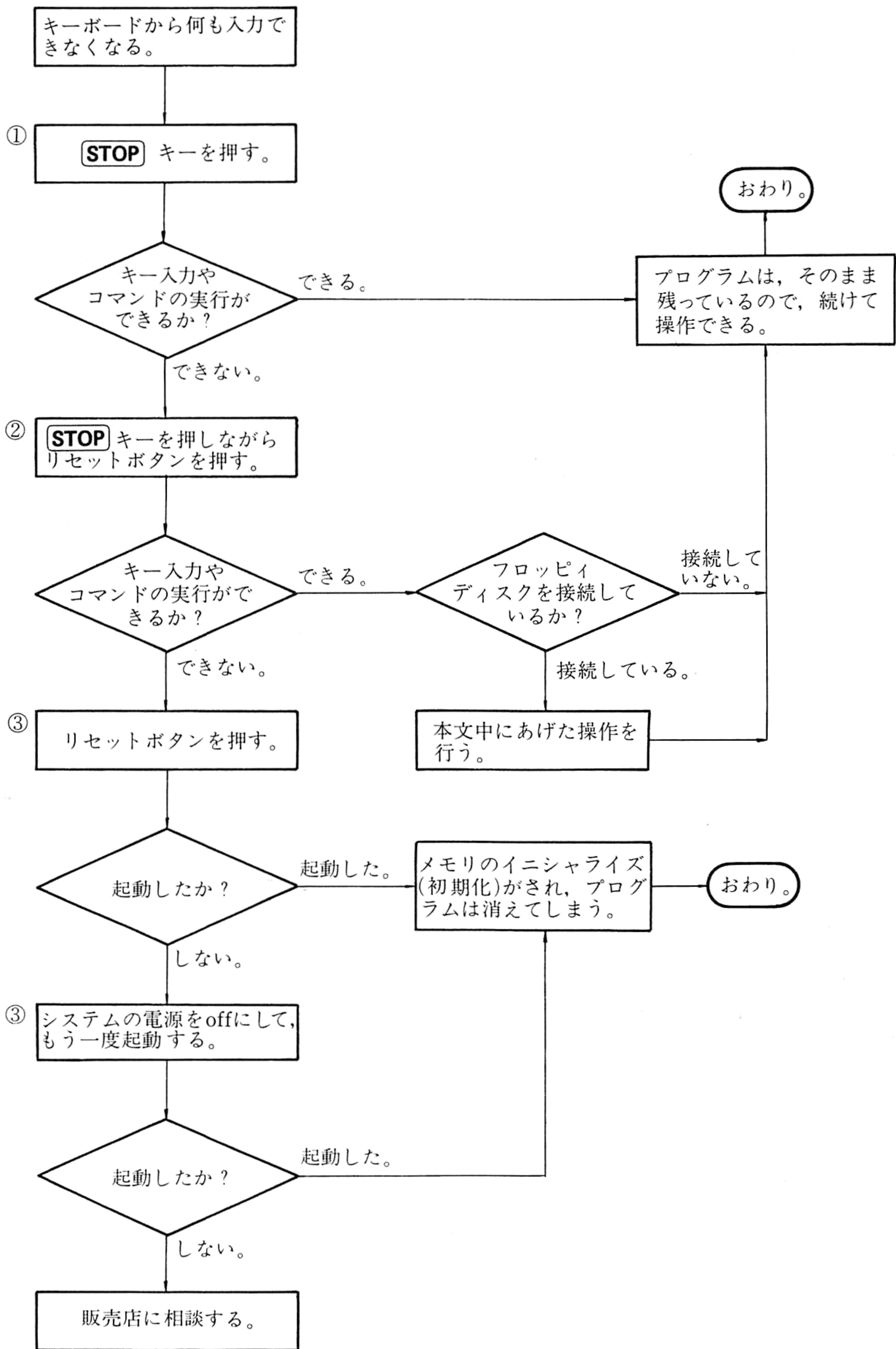



図３．５　キー入力できないときの操作

① **STOP キーを押す**

BASIC のプログラムが動いているときは，STOP キーを押すことによって中断し，コマンド入力待ち状態に戻すことができます。プログラムは，全くそのままの状態でメモリ上に残っています。ただしBASIC プログラムのなかで **out** や **poke** などを使っている場合はSTOP キーによって中断できなくなることがあります。

備　考 BASIC のプログラムを一時中断するときは，ESC キーを押します。続けるときは，任意の文字キーを押します。

② **STOP キーを押しながらリセットボタンを押す。**

STOP キーだけを押しても，まだキー入力ができない場合は，STOP キーを押しながらリセットボタンを押します。これは機械語のプログラムがでたらめに動いている場合の中断方法で，**ウォームスタート**（warm start）と呼ばれます。ウォームスタートが成功すると，画面には"**Ok**"と表示され，キー入力やコマンドの実行ができるようになります。

フロッピィディスクユニットを接続している場合は，ここで次のようにキー入力する必要があります。

N80DISK-BASIC の場合

```
CMD INIT RETURN
```

DISK-BASIC の場合

```
OUT &H31,1 RETURN
MON RETURN
G6005 RETURN
```

ウォームスタートを行ってもメモリ上のプログラムは，そのまま残っています。

③ **リセットボタンを押すか，電源を off にしてもう一度起動する**

ウォームスタートができない時には，メモリ上のプログラムを残すことをあきらめなければなりません。このようになるのは，コマンドの入力や実行ができない悲惨な状態にメモリが書き変えられてしまっ

たか，ハードウェアがでたらめに動き出してしまったかのどちらかが考えられます。

この場合は，リセットボタンを押すか電源を off にして,もう一度起動する（**コールドスタート**（cold start）といいます）しか手段がありません。コールドスタートをした場合メモリ上のプログラムは消えてしまいます。

リセットボタンを押しただけでは起動しない場合は，電源を off にしますが，このときは，5秒以上たってから，電源スイッチを on にし，リセットボタンを押してください。

3.8 電源を切る

電源を切ると，ユーザの入力したプログラムやデータはすべて消えてしまいます。電源を切るときは，以下のことを守ってください。

1. 重要なプログラムやデータを,カセットテープやフロッピィディスクにセーブし忘れていないか確かめます。
2. カセットテープやフロッピィディスクの読み書きの途中でないことを確かめます。読み書きの途中で電源を切ると、ファイルがこわれてしまいます。

（注　意）**フロッピィディスクユニットを使っている場合の特別な注意**

N80 DISK-BASIC では，オープンされたままのファイルがないことを確認してください。(念のためもう一度 CLOSEを実行してください)。 **DISK-BASIC**（**N-BASIC** モードのディスクバージョン）では，**REMOVE**したことを確認してください。

以上を確認したら，ドライブアクセス表示用 LED が消えていることを確かめたあと，フロッピィディスクをとり出してください(詳しくは8章で説明します。)。

これらをすべて確かめてから電源を off にします。電源を切る順序は，

MKII と周辺装置のどちらが早くてもかまいません。

電源を切った後，再度電源を入れる時は，５秒以上たってからにしてください。

第4章
キーボードの使い方とスクリーンエディタ

キーボードとディスプレイは，ユーザとMKIIの会話になくてはならない重要な入出力装置です。でたらめにキーボードを操作してもMKIIがこわれることはありませんから，いろいろ試して早く慣れてください。この章では，キーボードとディスプレイの使い方を簡単に紹介し，MKIIのもつ便利なスクリーンエディタについて説明します。

4.1　キーボードとディスプレイ

キーボードのキー配列はタイプライタに似ています。 MKII のキーボードはファンクションキーやテンキーなどをつけてパーソナルコンピュータとして使いやすく設計されています。

ディスプレイには，ブライト（明るさ）つまみがありますから見やすい状態に調整して使ってください。家庭用のテレビには，コントラストなどのつまみがあります。好みの状態に調整して使ってください。

図4．1　キーボードとディスプレイ

（PC-8001とくらべて） **MKII**のキーボードは **PC-8001** にはない TAB キーがついています。

4.2　キーボードの使い方

キーボードの使い方は，タイプライタとほとんど同じです。また，①同じキーを約1秒以上押し続けるとその文字を連続して入力できるオートリピート機能，②キーの内容を自由に指定できるファンクションキー，などの便利な機能が追加され使いやすくなっています。

以下に文字キーと特殊キーにわけて，キーボードの使い方を説明します。

4.2.1　文字キー

図4．2の，アミの部分が文字キーです。これらのキーを押すと，ディスプレイに，対応する文字が表示されます。

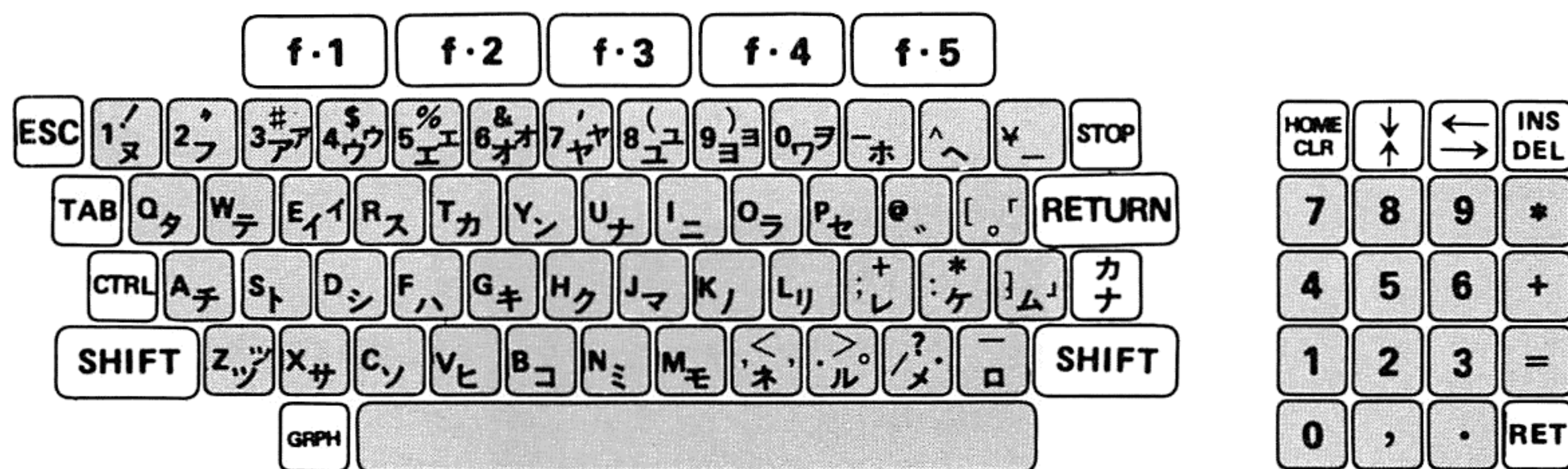



図4．2　文字キー

文字キーの入力方式には，ノーマルモード，グラフィックシンボルモード，カナモードの3つのモードがあります。

1．ノーマルモード

アルファベットや，数字，“+”，“*”，“?”などの特殊記号などの入力モードです。単独でキー入力した場合は，キートップの中段に書いてある文字が入力されます。アルファベットは，タイプラ

イタと同様に小文字が入力されます。SHIFTキーを押しながらキー入力した場合は，キートップの上段に書いてある文字または，アルファベットの大文字が入力されます。

[A/チ] =「a」

[SHIFT] + [A/チ] =「A」

[1/ヌ] =「1」

[SHIFT] + [1/ヌ] =「!」

図4．3　ノーマルモードのキー入力

2．カナモード

カタカナや“「”，“。”，“、”などの和文の特殊記号を入力するモードです。

[カナ]キーは，一度押すと，押したままの状態に固定されます。これを**ロックする**といいます。もう一度[カナ]キーを押すと，ロックが解除されます。

カナモードは，[カナ]キーをロックした状態でキー入力します。この場合 キートップの下段に書いてある文字が入力されます。さらにSHIFTキーを押しながらキー入力するとキートップの中段右側に書いてある文字が入力されます。

[カナ] + [>/。/ル] =「ル」

[SHIFT] + [カナ] + [>/。/ル] =「。」

図4．4　カナモードのキー入力

3．　グラフィックシンボルモード

図 4.5 のようなグラフィックシンボルや，“年”，“月”などの簡単な漢字を入力するモードです。GRPHキーを押しながら入力

します。グラフィックシンボルはキートップに書いてありませんから，図 4.5 や，付録1を見ながらキー入力してください。

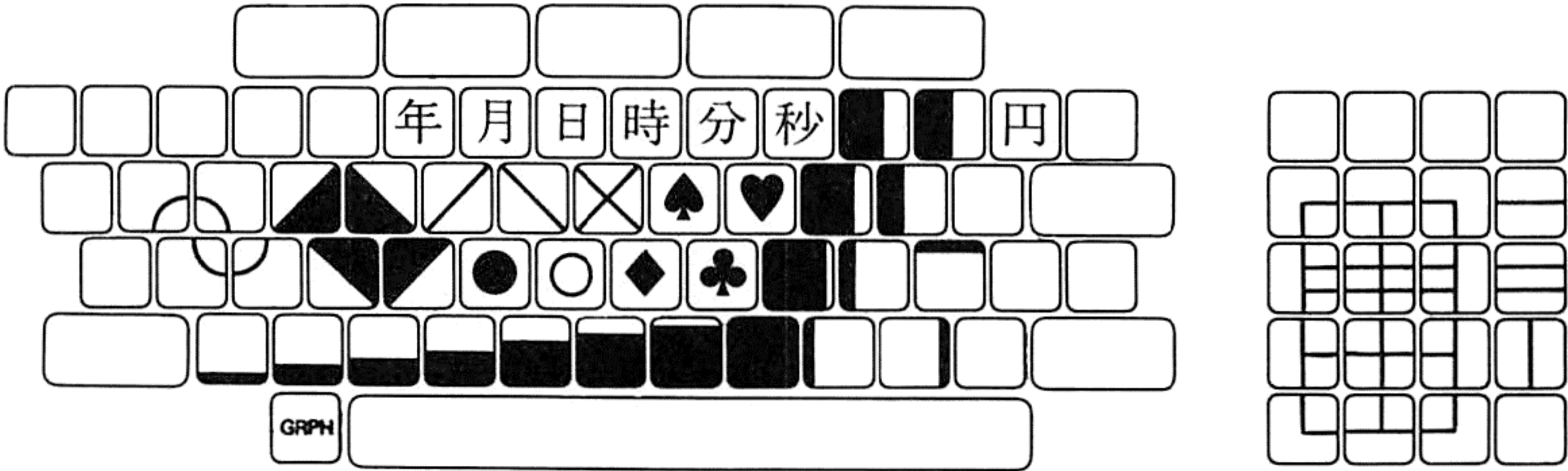



図4.5　グラフィックシンボルのキー配置

4.2.2 特殊キー

図 4.6 のアミの部分を特殊キーといいます。それぞれ特殊な働きをします。

図4.6　特殊キー

SHIFT　シフトキー

タイプライタのシフトキーと同じで，各キーのシフトポジションにある文字を入力する場合に使用します。また，アルファベットを入力する場合は，大文字の入力に使用します。

CTRL　コントロールキー

他のキーと組み合わせて，特殊なコードを入力します。

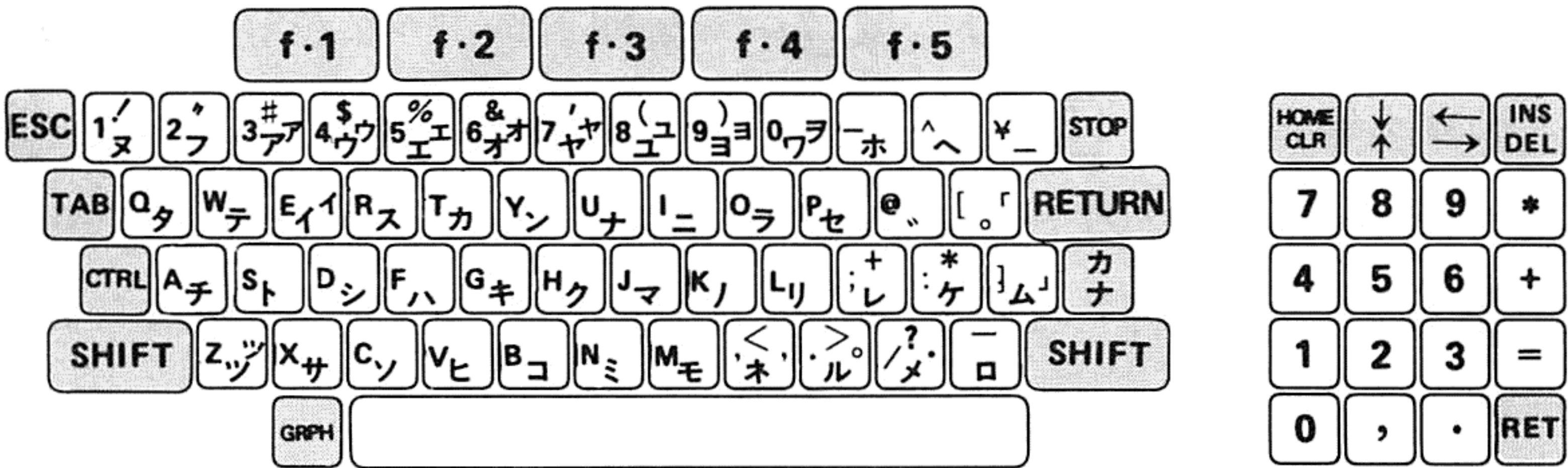



表4.1 CTRL キーの使い方

キー入力	動　　　　　作
CTRL + B	カーソルを1項目ごと左へ移動させます。
CTRL + E	カーソルの位置から，その行の終りまでをまっ消します。
CTRL + G	ブザーを鳴らします。
CTRL + H	1文字を削除します。 INS/DEL と同じ働きです。
CTRL + I	タブ位置までカーソルを移動させます。タブ位置は8桁ごとに設定されています。 TAB キーと同じ働きです。
CTRL + J	ラインフィード(LFコード)を挿入します。
CTRL + K	カーソルをホームポジション(左上スミ)に移動させます。SHIFT + HOME/CLR と同じです。
CTRL + L	テキスト画面をクリアします。HOME/CLR と同じです。
CTRL + M	RETURN と同じです。
CTRL + N	カーソルを1項目ごとに右へ移動させます。
CTRL + R	スペースを挿入します。SHIFT + INS/DEL と同じです。

RETURN , RET　リターンキー

このキーの入力により，1行の入力が終ります。このキーを入力するとカーソルは，次の行の先頭に移動します。

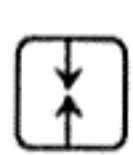
カナ　カナキー

このキーを押すと，キーはロックされ，入力はカナモードになります。 SHIFT キーを押しながら入力すると，キートップの中段の文字になります。

ロックされた カナ キーをもう一度押すと， カナ キーのロックは解除されます。

STOP　ストップキー

BASICのプログラムの実行を停止するときに使用します。

HOME/CLR　ホーム・クリアキー

このキーを入力すると画面がクリアされ，カーソルは画面の左上ス

ミに移ります。

[↕],[⇆]　カーソル移動キー

これらのキーと [SHIFT] キーの組み合わせでカーソルを上下左右に移動します。

[↕]　カーソルは1ポジション上へ動きます。カーソルが最上位にあるときは何もしません。

[SHIFT]+[↕]　カーソルは1ポジション下へ動きます。カーソルが最下位にあるときは一行スクロールします。

[⇆]　カーソルが1ポジション右へ動きます。カーソルが右端にあるときは次の行の左端に動きます。

[SHIFT]+[⇆]　カーソルは1ポジション左へ動きます。カーソルが左端にあるときは1行上の行の右端に移ります。

[INS DEL]　インサート／デリートキー

このキーを入力すると，点滅しているカーソルの左の文字が1文字消され,カーソルより右の1行分の文字列を左につめていきます。

[SHIFT]+[INS DEL]を入力すると，カーソルの前にスペースを1個挿入します。

[f・1]～[f・5]　ファンクションキー

[SHIFT] キーといっしょに使用する場合を含めて，10 種類の入力を行うファンクションキーとして使用できます。

最初は次のようにセットされています。

(注　意) N-BASIC モードでは [f・1] には [TAB] キーに相当するコードが入っています。

f・1	cmd	f・6	time$
f・2	auto	f・7	key
f・3	go to	f・8	print
f・4	list	f・9	list C_R
f・5	run C_R	f・10	cont C_R

(f・6～f・10は [SHIFT] キーを押しながら [f・1] ～ [f・5] を押します。)

このファンクションキーを押すと，定義されている文字例が入力されます。ですから，よく使う文字列をセットしておけば，プログラムやデータなどを入力する時間が，ずっと短くなります。各ファンクションキーは，最大15文字まで定義できますが，ディスプレイには，1行に40 文字を表示するときで5文字，80 文字を表示するときは，11 文字まで表示されます。

ファンクションキーを新しく定義する場合は，次のように入力します。

key 4, "run"+chr$(13) **RETURN**

これでf・4に，"run C_R" が定義されます。(**key** コマンドおよび **chr$** については「N80-BASIC リファレンスマニュアル」を参照してください。)

また，現在，定義されている各ファンクションキーの内容を表示させるには，次の命令を実行します。

key list **RETURN**

GRPH **グラフキー**

グラフィックシンボルを入力するときに使います。

ESC **エスケープキー**

プログラムの実行もしくは処理を一時停止させるときに使用します。再開は，任意のキーを入力します。

TAB **タブキー**（水平タブキー）

このキーを入力すると，カーソルは8文字単位で右へ移動します。

4.3 テキスト画面のディスプレイモード

キー入力された文字やプログラムの中でディスプレイに表示させた文字がかかれる仮想の黒板を，テキスト画面と呼びます。 **MKII** は，テキス

ト画面の他に**6章 グラフィックス**で述べるグラフィック画面を持っており，2枚の画面はそれぞれ独立しています。

この節では，テキスト画面のディスプレイモードについて説明します。ディスプレイの種類やプログラムによって，見やすいディスプレイモードを選んでください。

1．1行あたりの文字数，1画面あたりの行数のセット

MK IIを起動した時は，1画面に40文字×25行の文字を表示するモードにセットされています。これを **WIDTH** コマンドで設定しなおすことができます。

(書　式) **WIDTH** 〔<文字数／行>〕〔,<行数／画面>〕

文字数／行： 36, 40, 72, 80 のいずれか。

行数／画面： 20, 25 のいずれか。

(Sample) 80 文字×20 行に設定するとき

width 80, 20 RETURN

そのあと，80 文字× 25 行に直すとき

width , 25 RETURN

そのあと，72 文字× 25 行に直すとき

width 72 RETURN

このようにパラメータを省略すると，その部分はいままでの状態と同じ設定となります。

WIDTH コマンドで文字数／行，行数／画面をセットするとテキスト画面にキャラクタ座標が設定されます。座標は，左上が（0,0）右下が（文字数／行－1，行数／画面 －1）のようになります。例えば **WIDTH 80,20 RETURN** とした場合のキャラクタ座標は，図4.7 のようになります。

(注　意) カラーモードのときは，文字数／行は－1されます。例え

ば 80 文字／行と指定しても， 79 文字／行になります。

(0, 0)　(79, 0)
(0, 19)　(79, 19)

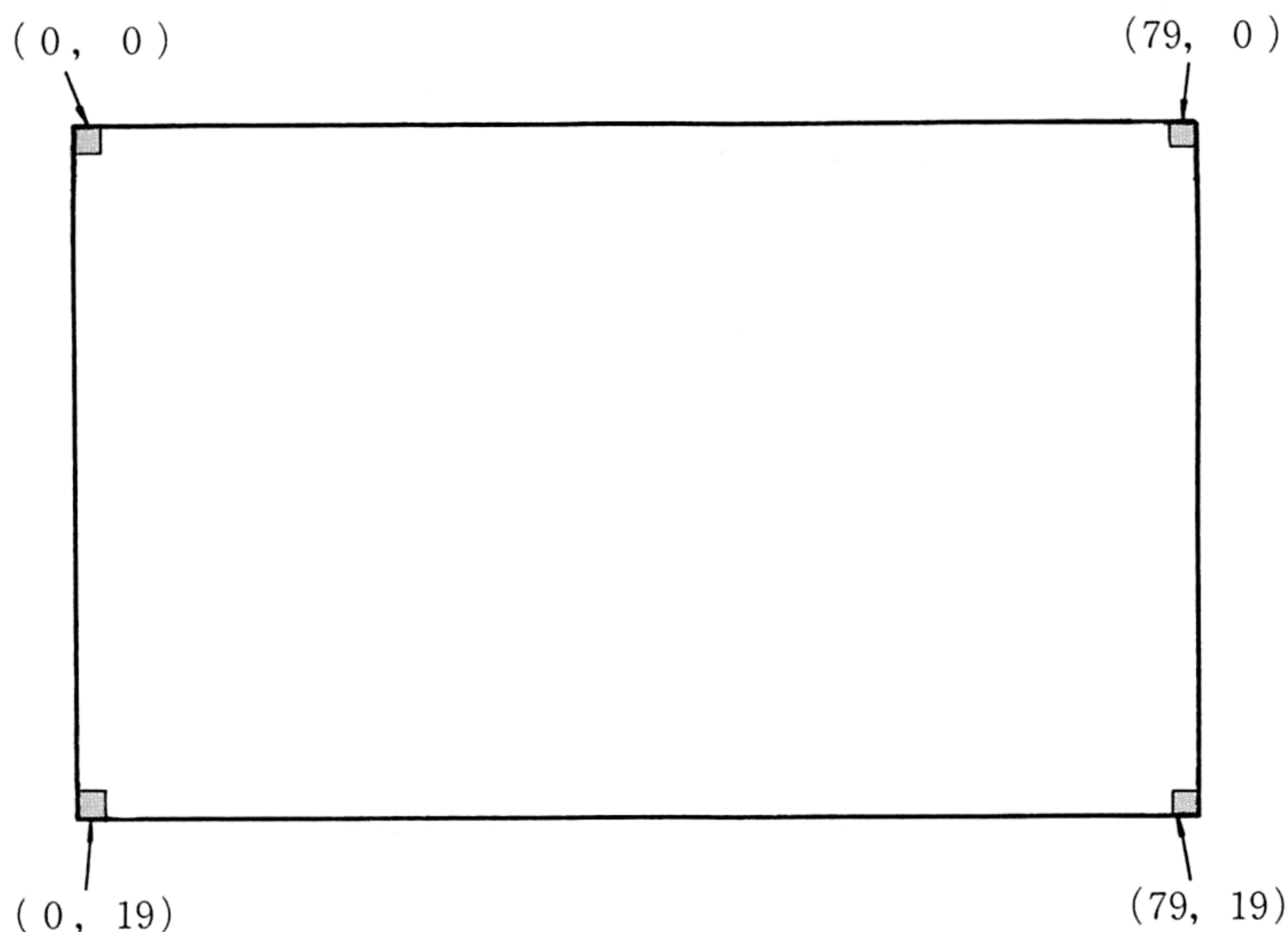



図4.7　テキスト画面のキャラクタ座標
(白黒モードで**WIDTH 80, 20**を設定したとき)

2．テキスト画面のモードセット

CONSOLE コマンドでセットします。

(書　式) **CONSOLE〔<スクロールのスタート行>〕〔,<スクロールする行数>〕〔,<ファンクションキーの表示の有無>〕〔,<カラーモードか白黒モードか>〕**

(1)　スクロールのスタート行と，スクロールする行数

WIDTH コマンドで設定した行数の最後まで文字を表示し，さらにもう一行表示する場合は，最上行が消え，一行ずつ上にずれます。これをスクロールといいます。

CONSOLEコマンドで，何行目から，何行をスクロールするかを指定できます。 **MK II** を起動した時は，最上行から最下行までをスクロールするように設定されています。 スクロールの範囲は **WIDTH** コマンドで設定した行数／画面の値以下でなければな

りません。

(Sample) 2行目から 10 行目までの9行をスクロールさせる場合

CONSOLE 1, 9 RETURN

このように **CONSOLE** コマンドも，パラメータを省略すると，それまでの状態と同じ設定となります。

(2) ファンクションキーの表示の有無

画面の最下行のファンクションキーの内容を表示するかしないかを設定できます。

表示するとき……… 1
表示しないとき…… 0

(Sample) ファンクションキーを表示させないように設定する。

CONSOLE , , 0 RETURN

(3) カラーモードか白黒モードかの選択

このパラメータは，次項の **COLOR** コマンドと組み合わせて画面に表示させるテキストの色や状態を設定するために使います。

カラーモードに設定するとき… 1
白黒モードに設定するとき …0

(Sample) 0行目から 25 行をスクロールし，ファンクションキーを表示させず，カラーモードに設定する場合

CONSOLE 0, 25, 0, 1 RETURN

(注　意) この章で扱うディスプレイモードは，すべてキャラクタ画面に関するモードです。第6章では，グラフィックスのモードが述べられていますが，混同しないようにしてください。

3．テキスト画面のカラー設定

COLOR コマンドでセットします。

(書　式) **COLOR〔<カラーコード>〕〔,<Nullキャラクタ>〕〔,<グラフィックモードかキャラクタモードか>〕**

(1) カラーコード

CONSOLE コマンドでカラーモードに設定した場合は，カラーコードで文字につける色を指定します。モノクロディスプレイを使っている場合には，色はつきませんが，明るさが変ります。カラーコードは 0～7 から選択します。

カラーコード	色
0	黒
1	青
2	赤
3	マゼンタ
4	緑
5	シアン
6	黄
7	白

CONSOLEコマンドで白黒モードに設定した場合は，カラーコードで文字の表示モードを指定します。

カラーコード	表示モード
0	ノーマル
1，3	シークレット(文字は表示されません)
2	ブリンク(文字が点滅します)
4	リバース
5，7	リバースシークレット
6	シークレットブリンク

注　意　カラーコマンドによって，表示させる文字の色や，表示モードを変えることができますが，一行の中では19回までしか色や表示モードを変えることはできません。

(2) Null キャラクタ

Null キャラクタとは，テキスト画面をクリアしたとき(HOME CLR キーを押したときまたは **CMD CLS 1** コマンドを実行したとき)テキ

スト画面をうめる文字のことです。**付録2 キャラクタコード表** のなかから選びます。

通常は，次項のパラメータでキャラクタモードを選んだときは20H（スペース），グラフィックモードを選んだときは，0H（ヌル）を選びます。

(Sample) **COLOR，&H20**

(3) グラフィックモードかキャラクタモードかの選択

MK II には，**PC-8001** の機能に，豊富なグラフィックスなどの機能が追加されています。**COLOR** コマンドによるグラフィックスモードは，**PC-8001** と同じグラフィックス機能を使う場合に選びます。このグラフィックスは，テキスト画面に書き込む方式になっています。グラフィックス機能については**第6章 グラフィックス**で，**PC-8001** のグラフィックス機能と，**MK II** のグラフィックス機能を比較して説明します。

4.4 スクリーンエディタ

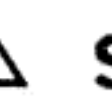
MK II は，ディスプレイを活用して，プログラムの作成および修正が，簡単に行えるように，「スクリーンエディタ」と呼ばれる画面編集機能を備えています。この機能により，修正したい行を改めて最初から入力する必要がなくなり，プログラムの入力や訂正も短時間で行えます。

(Sample 1) **一度入力した命令を修正してもう一度入力する**

まずキーボードから次のように入力します。

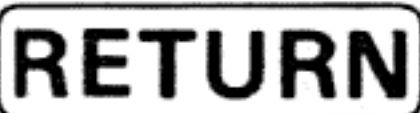
screen 3 [RETURN]

ブザーが鳴り画面には次のように表示されます。

```
screen 3
Syntax error
Ok
□
```

←これはカーソルです。

Syntax error は，「文法的にまちがっている」というエラーメッセージです。もう一度正しいコマンドを入力しなければなりません。このとき，**Ok** のあとに正しいコマンドを入力してもかまいませんが，一度入力したコマンドの一部を修正して入力し直す方が便利です。**screen 3**を **cmd screen 3** に修正します。このコマンドの意味は**第 6 章グラフィックス**でくわしく説明します。

screen 3の**s**の位置まで[↕]キーを使ってカーソルを移動します。

```
[s]creen 3
Syntax error
Ok
```

[SHIFT]キーを押しながら[INS DEL]キーを4回押して，スペースを4文字分あけます。

```
[△]△△△ screen 3
```

キーボードから **cmd** と入力します。

```
cmd □ screen 3
```

これで修正ができたので，この位置で[RETURN]キーを入力します。[RETURN]キーは，**MK II**にコマンドを入力するのに使います。[RETURN]キーは，**cmd screen 3**と表示されている行のどの位置で入力してもかまいません。

[RETURN]キーを入力すると，コマンドが実行されます。コマンドの実行が終わると“**Ok**”と表示されます。

```
cmd screen 3
Ok  tax error
□
```

これはゴミです。

このように，画面に表示されているコマンドは，修正して何度でも実行させることができます。修正しなくてもコマンドの表示されている行にカーソルを移動して[RETURN]キーを入力すれば何回でも実行することがで

きます。

(Sample 1)のように，プログラム中ではなく，単独にコマンドを実行させるモードをダイレクトモードといいます。ダイレクトモードで入力したコマンドは，スクロールして画面に表示されなくなってしまったら，スクリーンエディットすることができません。これに対してプログラムモードでは，何度でもプログラムを表示させて，修正することができます。

(Sample 2) **プログラムを修正する**

まず，プログラムを作ります。次のようにキーインしてください。

```
10 cmd screen 2, 0, 2 [RETURN]
20 code circle (160, 100) 70, 3 [RETURN]
[f・4] [RETURN]
```

[f・4] は，“**list**”とセットされています。

LISTコマンドは，プログラムリストを表示します。

プログラムモード（行番号をつけてコマンドなどを入力するとき）は [RETURN] キーを入力しても，コマンドは実行されません。プログラムを実行させるときは **RUN**コマンドを使います。[f・5] は“**run** [RETURN]”とセットされています。[f・5] を押すと，次のように表示されます。

```
run
Syntax error in 20
Ok
□
```

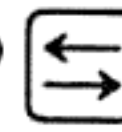
20行に文法的な誤りがあるというメッセージが出ました。したがって 20行を修正しなければなりません。**LIST**コマンドでプログラムのリストを表示し，カーソルを修正したい箇所まで移動して，文字の挿入，削除，変更などを行います。[f・4] [RETURN] と押すと次のように表示されます。

```
list
10 CMD SCREEN 2, 0, 2
```

20 CODE CIRCLE (160, 100)70, 3
Ok
□

20 CODE CIRCLE(160,100)70, 3 を **20 CMD CIRCLE (160, 100), 70,3**に修正します。大文字でも小文字でもかまいません。

カーソルを[↕]キーと[⇆]キーを使って**O**の文字の位置まで移動し，[M]キーを押します。

20 Cm[D]E CIRCLE (160, 100)70, 3

カーソルを削除する**E**の右どなりの位置まで移動し，[INS DEL]キーを押します。

20 CmD□CIRCLE (160, 100)70, 3

カーソルを7の文字の位置まで移動し，[SHIFT]キーを押しながら，[INS DEL]キーを押して「,」を入れるスペースをあけ，[< ,]キーを押します。

20 CmD CIRCLE (160, 100),[7]0, 3

修正が終わったので[RETURN]キーを押します。[RETURN]キーを押すことによってその行が入力されます。もう一度**LIST**コマンドでプログラムを表示して，修正ができたかどうか確かめてください。

2つの例で，スクリーンエディタの使い方をご紹介しました。例では[↕]キーや[⇆]キーでカーソルを移動しましたが，カーソルを移動するには，他にも，表 4.1のような便利な方法があります。これらを上手に使って，能率的にスクリーンエディットしてください。

(注　意) スクリーンエディタ注意事項

(1) [RETURN]キーを押さないと何も修正されません。

(2) 行番号の部分を変えて[RETURN]キーを押すと，新しい行ができますが，古いものもそのまま残ります。

(3) 行を消すには，行番号だけ入力して[RETURN]キーを押し

ます。

(Sample) 20行を消すとき

20 RETURN

(参　照) たくさんの行を消すときは，**DELETE**コマンドを使うのが便利です。「N80-BASIC リファレンスマニュアル」をご覧ください。

(4) 一行は，248文字までの長さで作れます。2行以上にわたる場合は，行が変わるところで RETURN キーを押してはいけません。 248文字までの行を入力して， 一回だけ RETURN キーを押せばよいのです。

(5) プログラムが大きいときは， **LIST**<何行目から>－<何行目まで>で，範囲を指定して，表示させることができます。また，**LIST.** で，最後に参照した行だけを表示できます。

(Sample 1) 200行め以降を表示させる

LIST 200－

(Sample 2) 最後に参照した行から， 150行までを表示させる

LIST. －150

第5章
ソフトウェア概説

この章では，MK II のもつ2つのプログラミング言語 N_{80}-BASIC, N-BASIC などのソフトウェアについての概要を述べます。

5.1 N_{80}-BASICとN-BASICの包含関係

MK II は，N_{80}-BASIC と N-BASIC という2つの BASIC インタプリタ* を持っています。このうちの N-BASIC は，PC-8001 に搭載されているプログラミング言語ですから，MKII を N-BASIC モードに設定しておけば，BASIC だけを使う場合，PC-8001 と全く同じように使うことができます。

MKII のハードウェアを生かした N_{80}-BASIC は，N-BASIC と非常に密接な関係にあります。N_{80}-BASIC と N-BASIC は，別々のものではなく，N-BASIC が N_{80}-BASIC の一部になっているのです。

これはメモリマップを使って説明するとよくわかります（図5.1）。

MK IIは，ROM**(BASIC を動かすプログラムが書き込まれたメモリ）を 32K バイト持っています。N_{80}-BASIC を起動すると，この32K バイトすべてが使われます。N-BASIC を起動すると，24K バイトが使われ，残りの 8K バイトはユーザが使うことができます。言いかえれば，N_{80}-BASIC は，N-BASIC に 8K バイト追加してできているのです。

---

* インタプリタ (interpreter)
　　プログラムを1命令ずつ翻訳しては，その解釈通りに実行するもの。

** ROM (Read Only Memory)
　　読み出し専用メモリ。ROM の内容を変えることはできません。

Address
7FFFH
追加された
8Kバイト
6000H
5FFFH
N$_{80}$-BASIC
32Kバイト
N-BASIC
24Kバイト
0000H

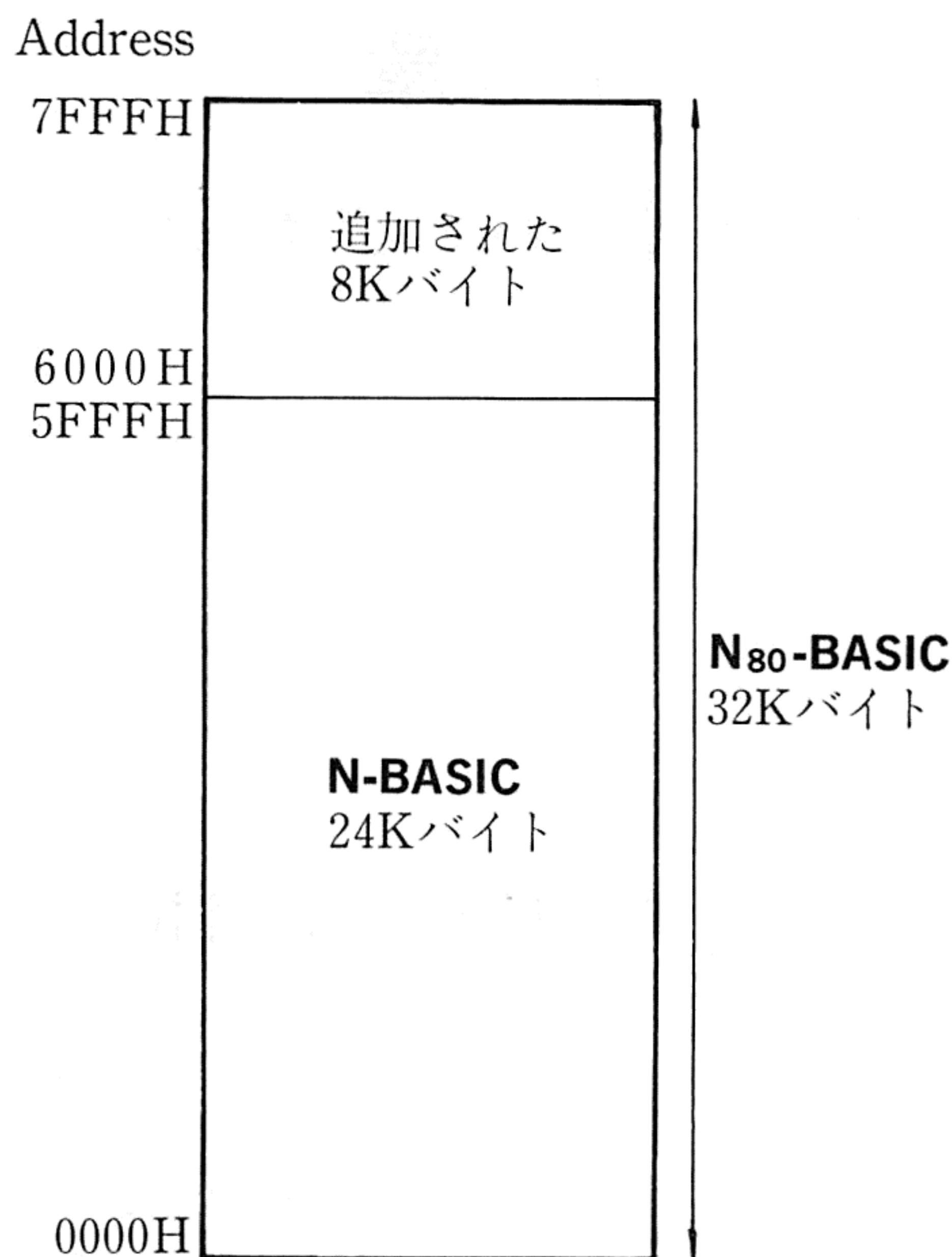



図5.1　**ROM**のメモリマップ

(参　照) メモリマップは，**第12章　メモリマップ**で説明します。**ROM**については，**第7章　カセットテープの使い方**で詳しく説明します。

従って，**N$_{80}$-BASIC** は，**N-BASIC** のすべての機能を持っています。ですから，**N-BASIC** で書いてあるプログラムはすべて **N$_{80}$-BASIC** で動かすことができるわけです。

(注　意)(PC-8001とくらべて) ただし，**PC-8001** 用のソフトウエアでも機械語で書かれたもののなかには**MKII** で使えないものもあります（BASICで**POKE, USR**などを使っているプログラムも含みます)。これは**PC-8001**と**MKII** では割り込みの優先順位が一部ちがうためです。割り込みについては第14章をごらんください。

N-BASIC になく，**N$_{80}$-BASIC** が持っている機能は，次の3つに大別されます。

1．豊富なグラフィックス機能

PC-8801 の **N$_{88}$-BASIC** のもっているグラフィックス機能のなかから選んで **N$_{80}$-BASIC** に盛り込みました。

(参　照) **第6章　グラフィックス** に詳しく説明されています。

2．8インチフロッピィディスクをサポートする

N$_{80}$DISK-BASIC では，ミニフロッピィディスクユニットに加えて，8インチ標準フロッピィディスクユニットを接続することができます。また機械語のプログラムのロード，セーブもできます。

また，**MOUNT** コマンド，**REMOVE** コマンドが不要になりました。

(参　照) **第8章　フロッピィディスクの使い方** で詳しく述べます。

3．漢字が使える

オプションの漢字ボード（**PC-8001MKII-01**）を接続すれば漢字が使えるようになります。

(参　照) **第15章　漢字の使い方** に詳しく説明されています。

5.2　N$_{80}$-BASICとN-BASICの切り換え

N$_{80}$-BASIC と **N-BASIC** は,それぞれ表5．1にあげる方法によって起動します。

表5．1　**N$_{80}$-BASIC** と **N-BASIC** の切り換え方法

ディップスイッチ8	起　動　方　法	選択されるBASIC
off	電源を入れる，または リセットスイッチを押す。	N$_{80}$-BASIC
on		N-BASIC
off	CTRL キーを押しながら 電源を入れる。または CTRL キーを押しながら リセットスイッチを押す。	N-BASIC
on		N$_{80}$-BASIC

つまり，どちらの BASIC が選ばれるかは，ディップスイッチ8と，起動のときに CTRL キーを使うか使わないかによって決まります。

CTRL キーを押した場合は，押さなかったときと逆になります。

5.3 MKIIのモード各種

MKIIは，BASIC モードの他にターミナルモード，モニタモードを持っています。

表5.2 **MKII**のモード各種

BASICモード	N_{80}-BASIC　N_{80} DISK-BASIC N-BASIC　DISK-BASIC
ターミナルモード	
モニタモード	

BASIC には，4つのモードがあります。2つの ROM-BASIC モード（フロッピィディスクユニットを接続しないときのモード）と，2つの DISK-BASIC モードです。4つの BASIC モードの機能の包含関係を図5.2 に示します。

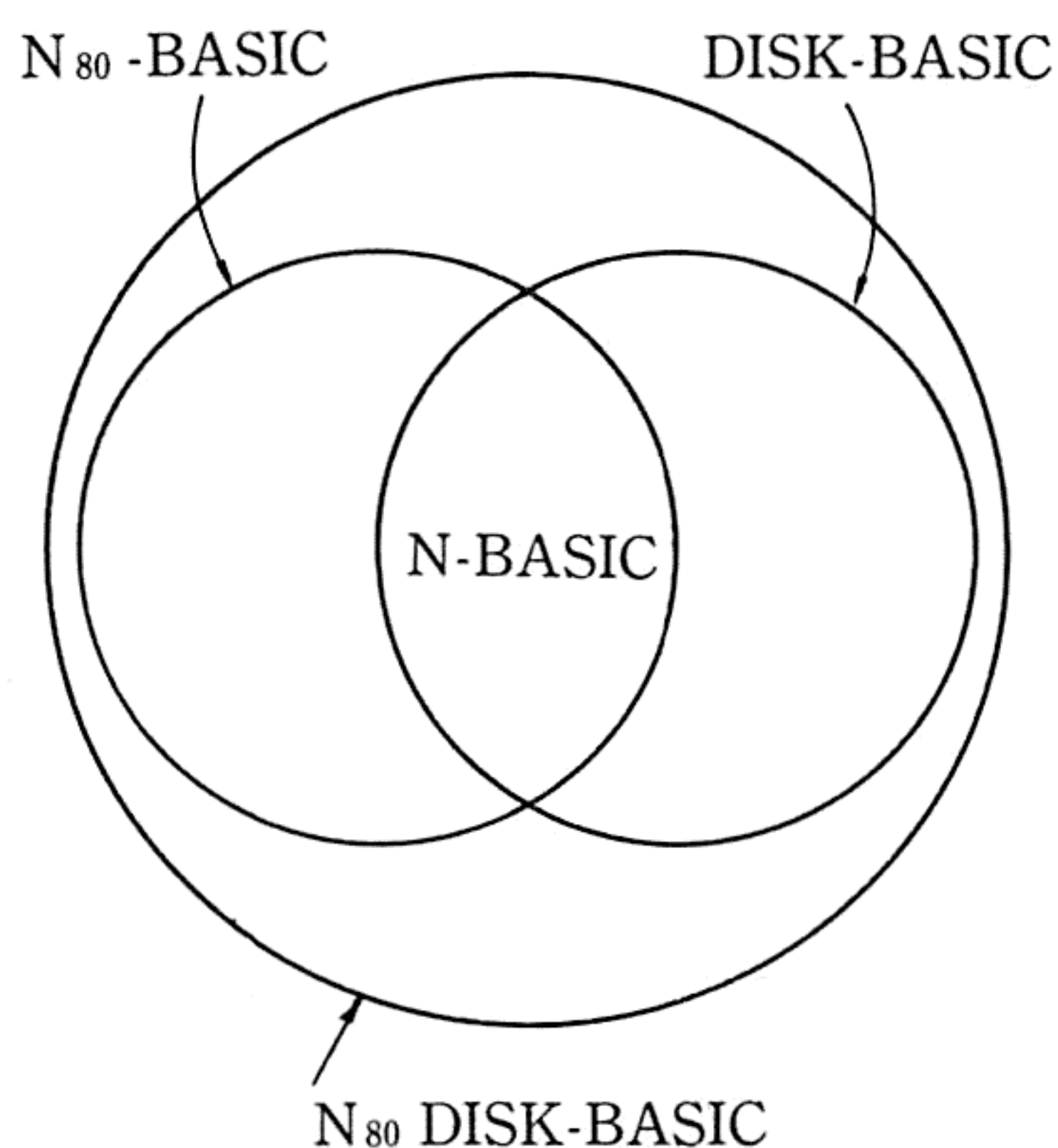



図5.2 4つのBASICモードの機能包含関係

ターミナルモードやモニタモードに入るには，BASIC モードからそれぞれ**TERM**コマンド，**MON**コマンドを使います。

(参　照) **第9章 ターミナルモード，第10章 モニタ**

5.4 ソフトウェアを使う上での一般的注意事項

1．**PC-8001 用のプログラムを使用するときは，**ディップスイッチ 8 を on にセットして，**N-BASIC モードで使用してください。**N_{80}-BASIC モードで使用すると，ほとんどの場合は正常に動くのですが，次のような場合は正常に動作しません。

① 機械語のサブルーチンなどが，**N_{80}-BASIC** のワークエリアを使用している。

② プログラムや変数領域が大きくて，実行すると Out of Memory エラーが生ずる。

これは，**N-BASIC** と，**N_{80}-BASIC** のワークエリアが異なるためです。

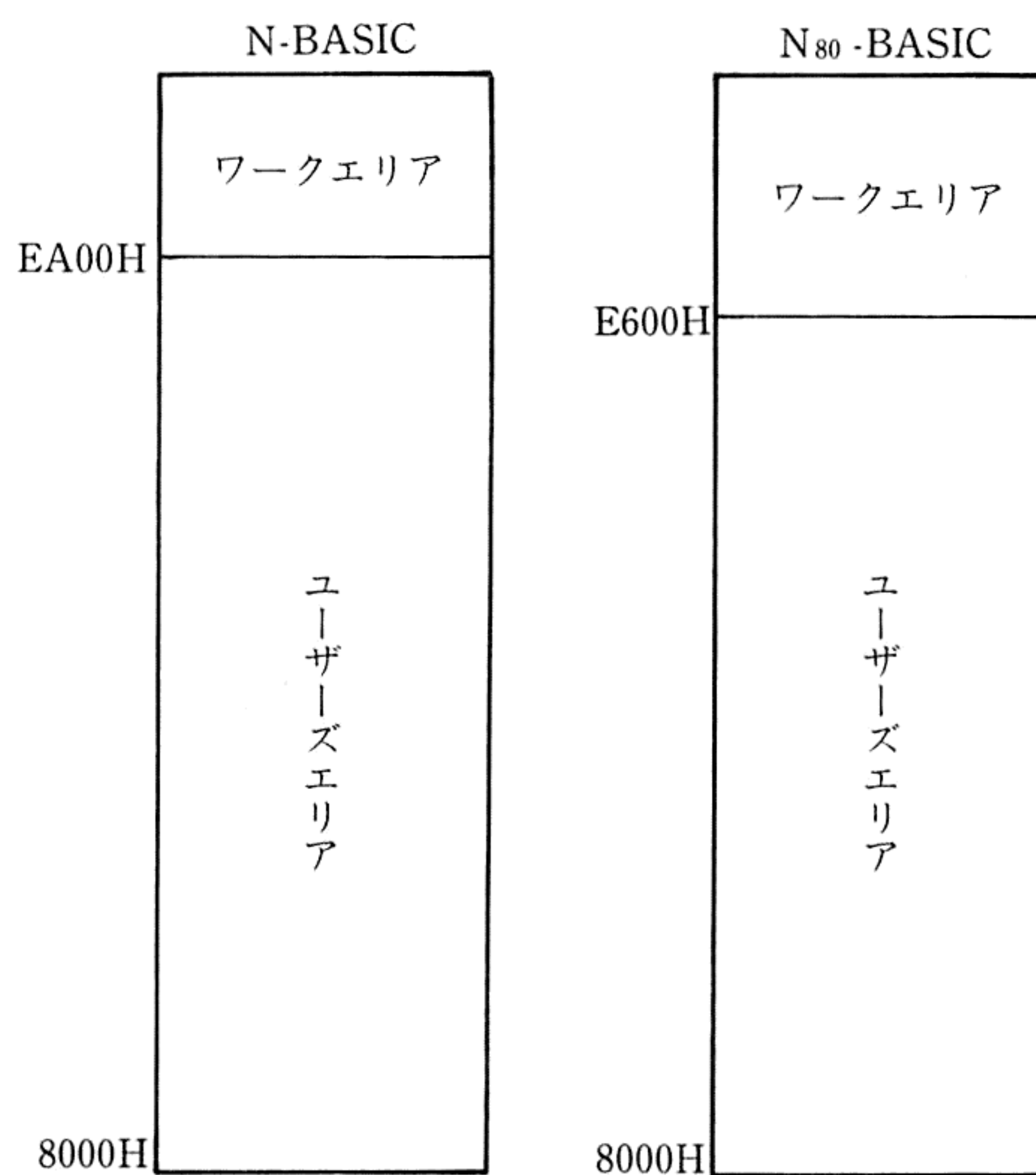



図5．3　ワークエリアのちがい（ROM-BASICモードの場合）

図5.3はフロッピィディスクを接続していないシステム(ROM-BASIC モード)でのメモリマップです。**N_{80}-BASIC** のワークエリアは **N-BASIC** のそれより 1K バイト大きくなっています。つまりユーザーズエリアが 1K バイト小さくなっているわけです。したがって，PC-8001 用のプログラムで E600H 番地から EA00H番地を使っているプログラムを **N_{80}-BASIC**上で動かすことはできないのです。

(参 照) **第 10 章 モニタ**

2. **MK II** と**PC-8001**では割り込みの優先順位がちがうので**PC-8001**用のプログラムのうち割り込みを使っているものは**MKII**を **N-BASIC**モードにしても使えないことがあります。

(参 照) **第 14 章 割り込み**

3. **N_{80}-BASIC, N_{80}DISK-BASIC** において増設 ROM の 6000H 番地から 7FFFH 番地に書かれた機械語プログラムを **USR** 関数で呼び，その値を **CMD** のついた命令や **STATUS** のついた関数の中で使うような場合は，①のようにします。②のように直接使うことはできません。

① 正しい例

```
10 DEFUSR=&H6808
15 TMP= USR(0)
20 CMD VIEW(TMP, 1)-(100, 100)
```

② 誤っている例

```
10 DEFUSR=&H6808
20 CMD VIEW(USR(0), 1)-(100, 100)
```

4. **PC-8001**用の増設ROMは，ピンコネクションがちがうため，**MKII**の増設ROMソケットにそのままさして使うことはできません。**12.1 増設ROMの使い方** に述べる2764タイプのPROMに書き移せば使えます。

第6章
グラフィックス

N_{80}-BASICとN-BASICの最も大きな違いはグラフィックス機能である，といっても過言ではありません。N-BASICにはなく，N_{80}-BASIC だけが持っている命令にはCMD が，関数にはSTATUSがついていますが，これらの命令や関数のほとんどがグラフィックスに使われます(図6．1参照)。

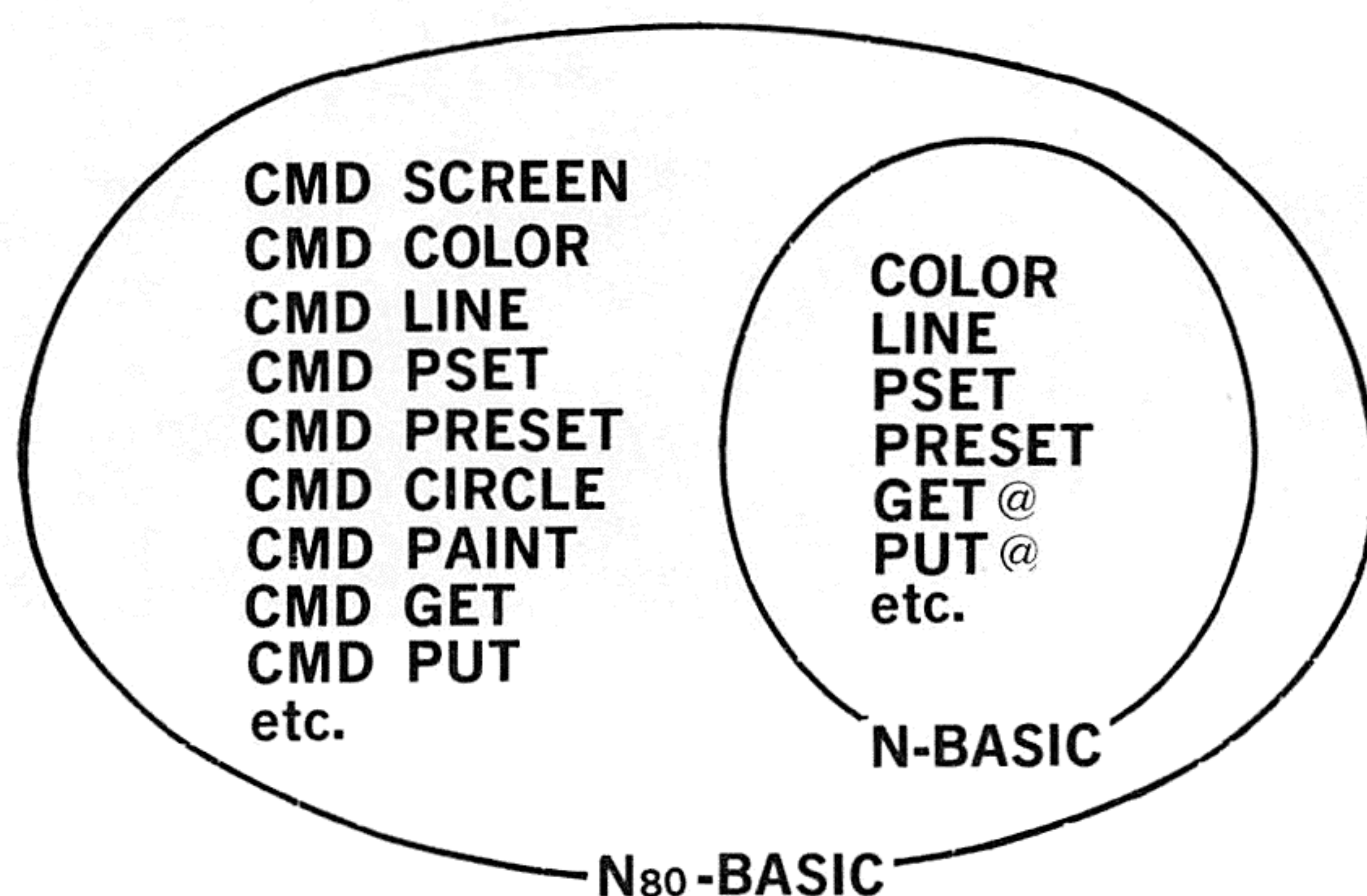



図6．1　N_{80}-BASIC と N-BASICのグラフィックス機能

この章では，おもに図 6.1で CMD のついている命令で扱うグラフィックスについて述べます。 N-BASIC のグラフィックスについては6．11節で述べます。

6.1 N-BASICのグラフィックスと比較して

次の例のように，キー入力して実際に画面に絵をかいてみてください。

(Sample)

```
width 40, 25 [RETURN]
console 0, 25, 0, 1 [RETURN]
color 6, 0, 1 [RETURN]
```

} N-BASIC のグラフィックスのモード設定

```
line (0, 0)-(79, 99), pset [RETURN]
cmd screen 2, 0, 5 [RETURN]   ┐ 高解像度グラフィックスの
cmd color 3, 0 [RETURN]       ┘ モード設定
cmd cls 2 [RETURN]
cmd line (319, 0)-(0, 199), 3 [RETURN]
```

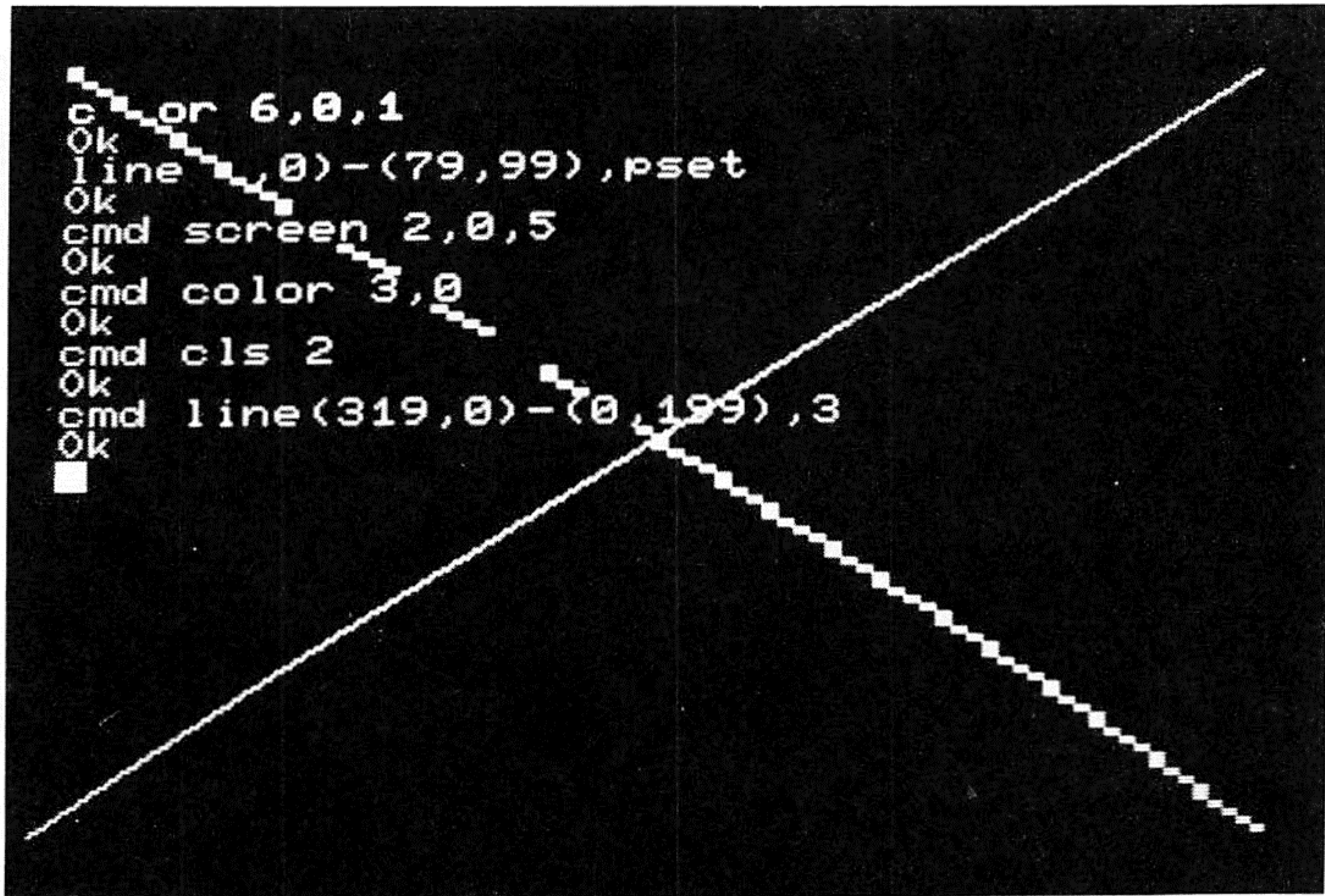



図６．２　**N-BASIC** のグラフィックスと比較して

この例は，**N-BASIC** のグラフィックスと，**CMD** 命令で動く **N$_{80}$-BASIC**のグラフィックスを，一諸に表示させたものです。これでわかるように，２つのグラフィックスは，解像度がちがいます。また，**SHIFT** キーを押しながら何回か ↕ キーを押すと，**N-BASIC** のグラフィックスはテキストと一緒にスクロールして消えてしまいますが，**CMD** 命令でかいたグラフィックスは，テキスト画面に関係なく，画面上にそのまま残っている，というちがいがあります。

このような２つのグラフィックスを次のように区別することにします。

N-BASIC の グラフィックス	―	**低解像度**(Low Resolution) **グラフィックス** グラフィック座標を（Lx，Ly）で表わす。
CMD 命令でかく グラフィックス	―	**高解像度**(High Resolution)**グラフィックス** グラフィック座標を（Hx，Hy）で表わす。

低解像度グラフィックスについては，６．11 節で述べます。次節から

は，高解像度グラフィックスについての説明をしていきます。

6.2　テキスト画面とグラフィックス画面

テキスト画面については，4．3節で詳しく述べました。

MKIIが ディスプレイに描く画面には，テキスト画面とグラフィックス画面との2枚があります。この2枚の画面は，図 6.3 のように透明のシートに描かれて，重なっているように見えます。

ユーザは，この2枚の画面を，別々に操ることができます。

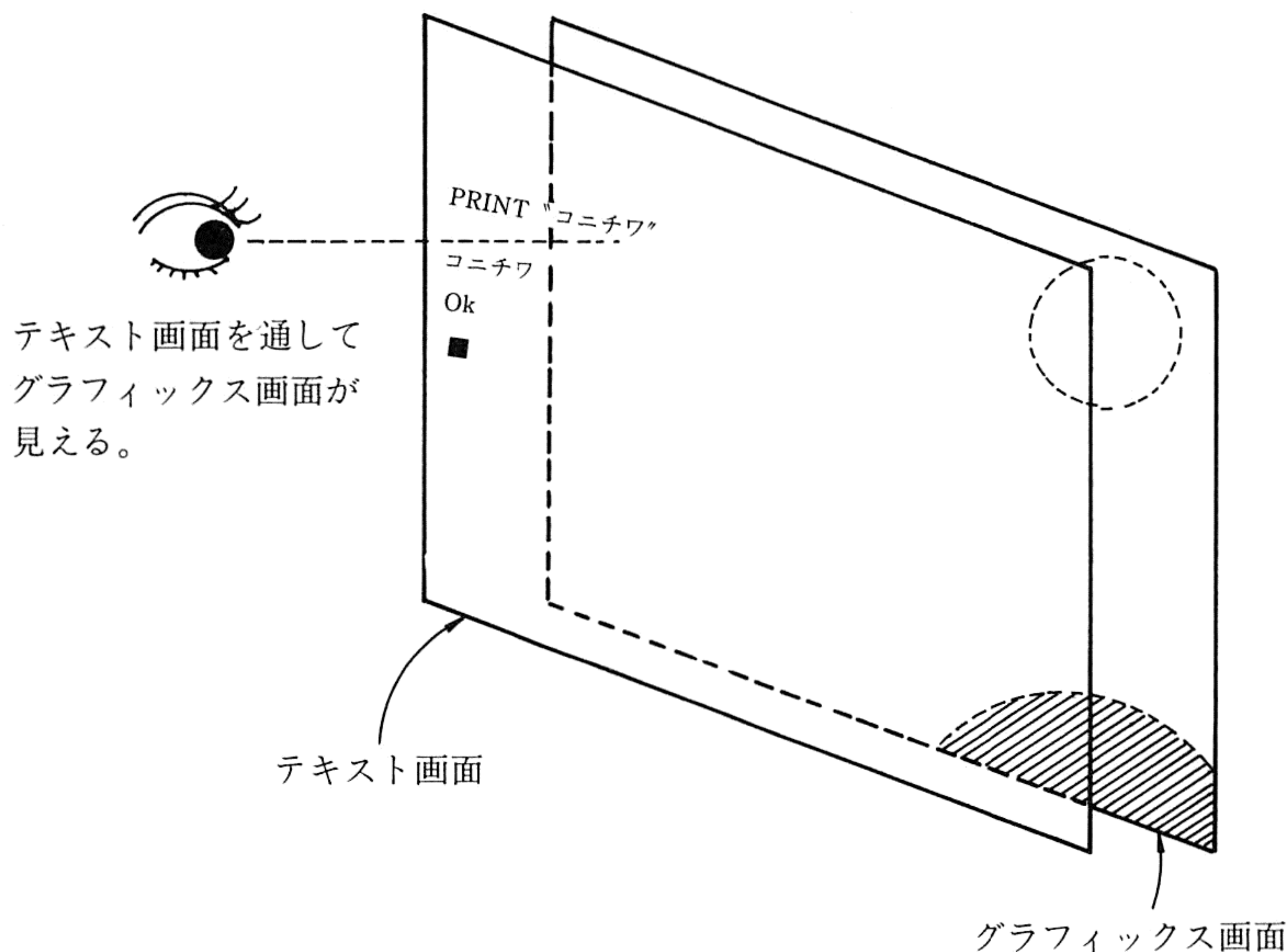



図6．3　テキスト画面とグラフィックス画面

6.3　画面とVRAM（Video RAM）

ディスプレイ画面用の情報が書き込まれるメモリを**VRAM**（Video

RAM）とよんでいます。 **MKII** は，VRAM のエリアを２つ持っています。１つは，テキスト画面用，もうひとつは，グラフィックス画面用です。２つの VRAM が独立しているため，２枚の画面を別々に扱えるわけです。

（PC-8001とくらべて） **PC-8001** では，図６．２ の例のように グラフィックス専用のVRAMはなく，ひとつのVRAMをテキストとグラフィックスで兼用していました。したがって，グラフィックスも，テキストと一緒に扱わなければならず，解像度も低いものでした。

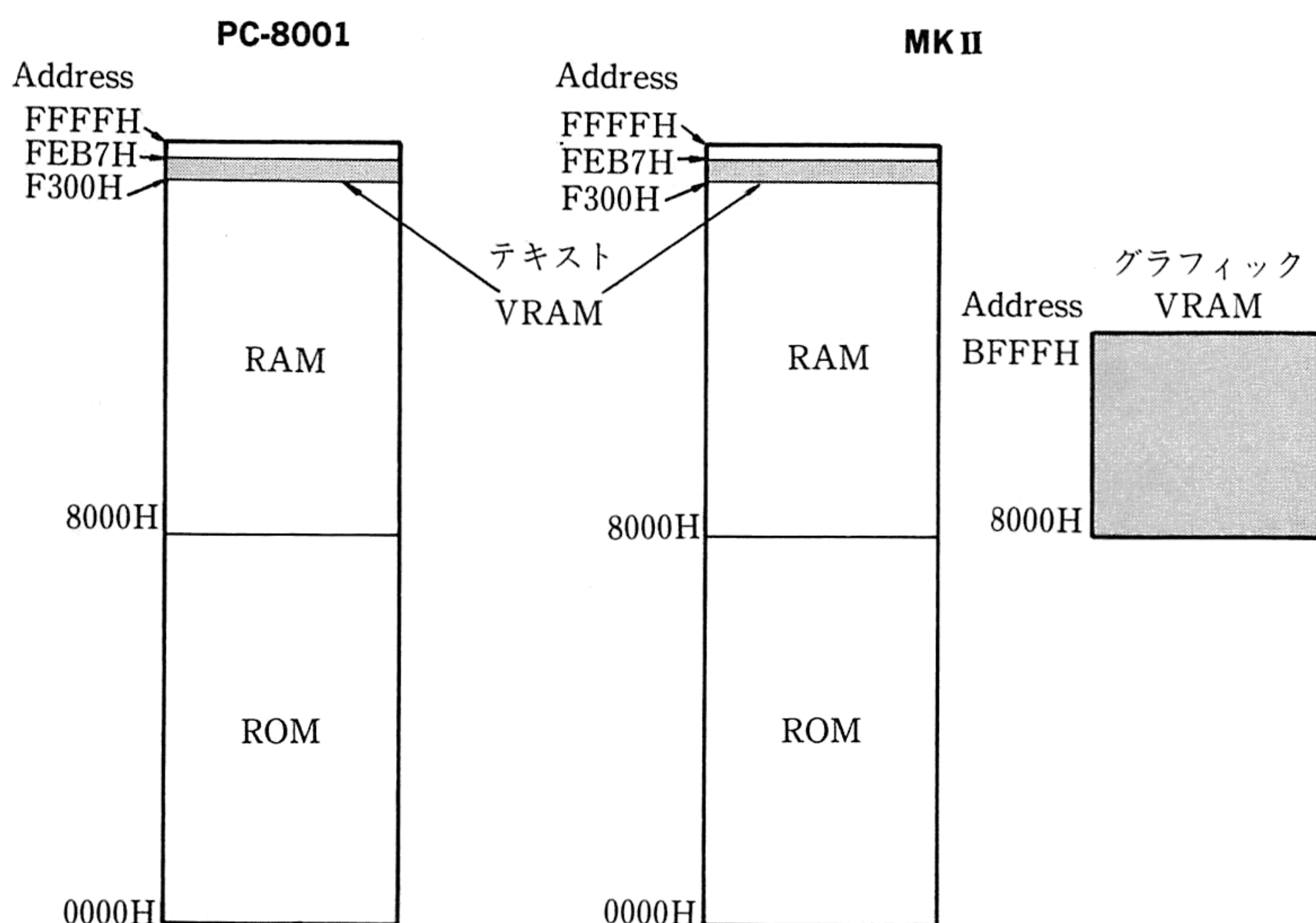



図６．４　**PC-8001**と **MKII** とのVRAMのちがい

6.4　グラフィック座標系

6.4.1　ピクセル(pixel)

画素と訳します。ディスプレイに表示されるひとつひとつの点のことを**ピクセル**と呼びます。正確には，独立に，制御できる最小の単位です。

6.4.2 解像度の異なる2つのグラフィック座標

テキスト画面に，キャラクタ座標が設定されているように，グラフィック画面にもグラフィック座標があり，ひとつひとつのピクセルをX座標の値とY座標の値で表わします。

グラフィックスの座標系には，1画面 640×200 ピクセルの座標系と，1画面320×200ピクセルの座標系の2種類があります。この座標系の設定は，**CMD SCREEN**によって行います。詳しくは6.8節で述べます。

ピクセルに対応する座標値は，図 6.5 のようになります。

(0, 0)　(639, 0)

(0, 199)　(639, 199)

640×200ピクセル

(0, 0)　(319, 0)

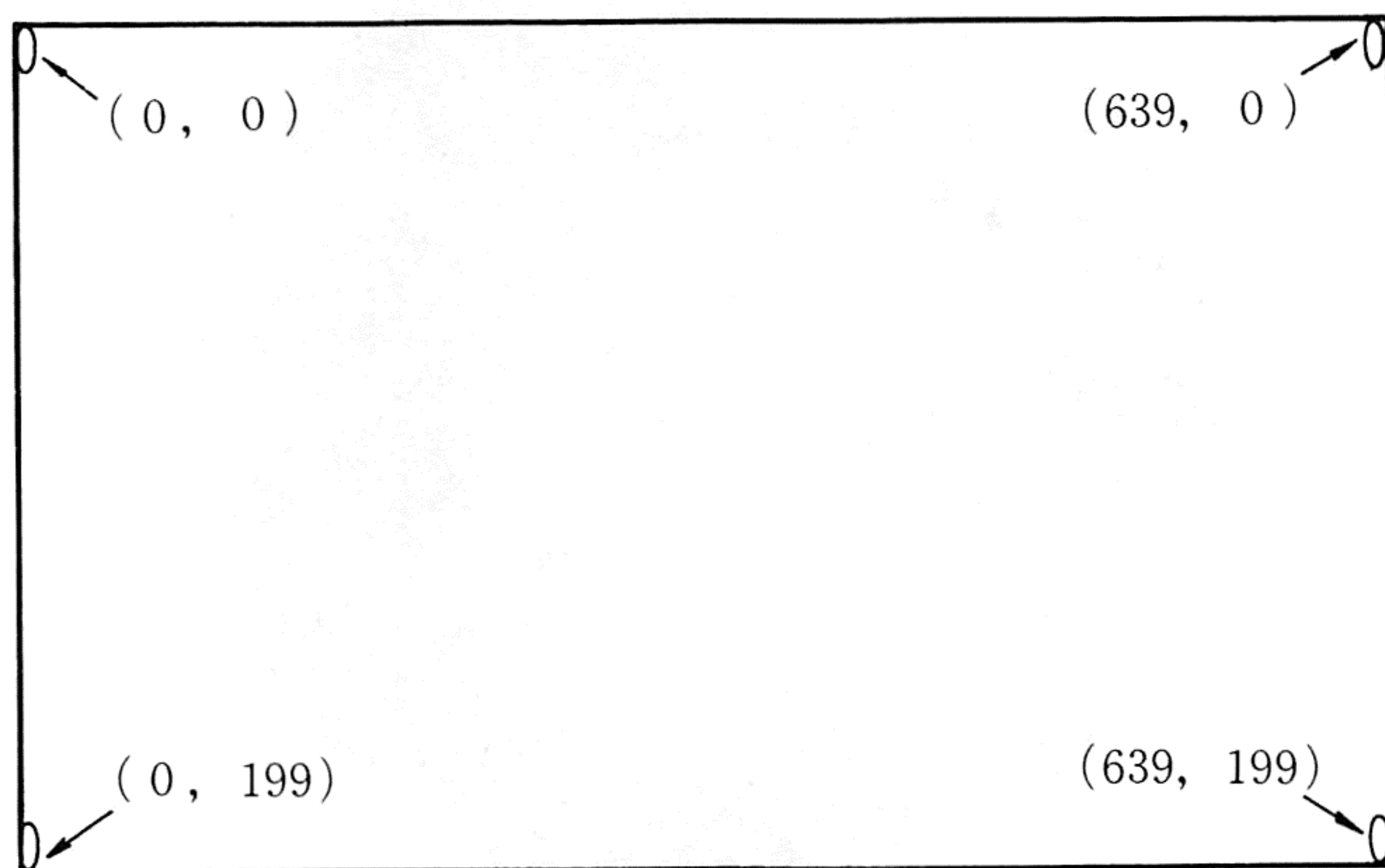
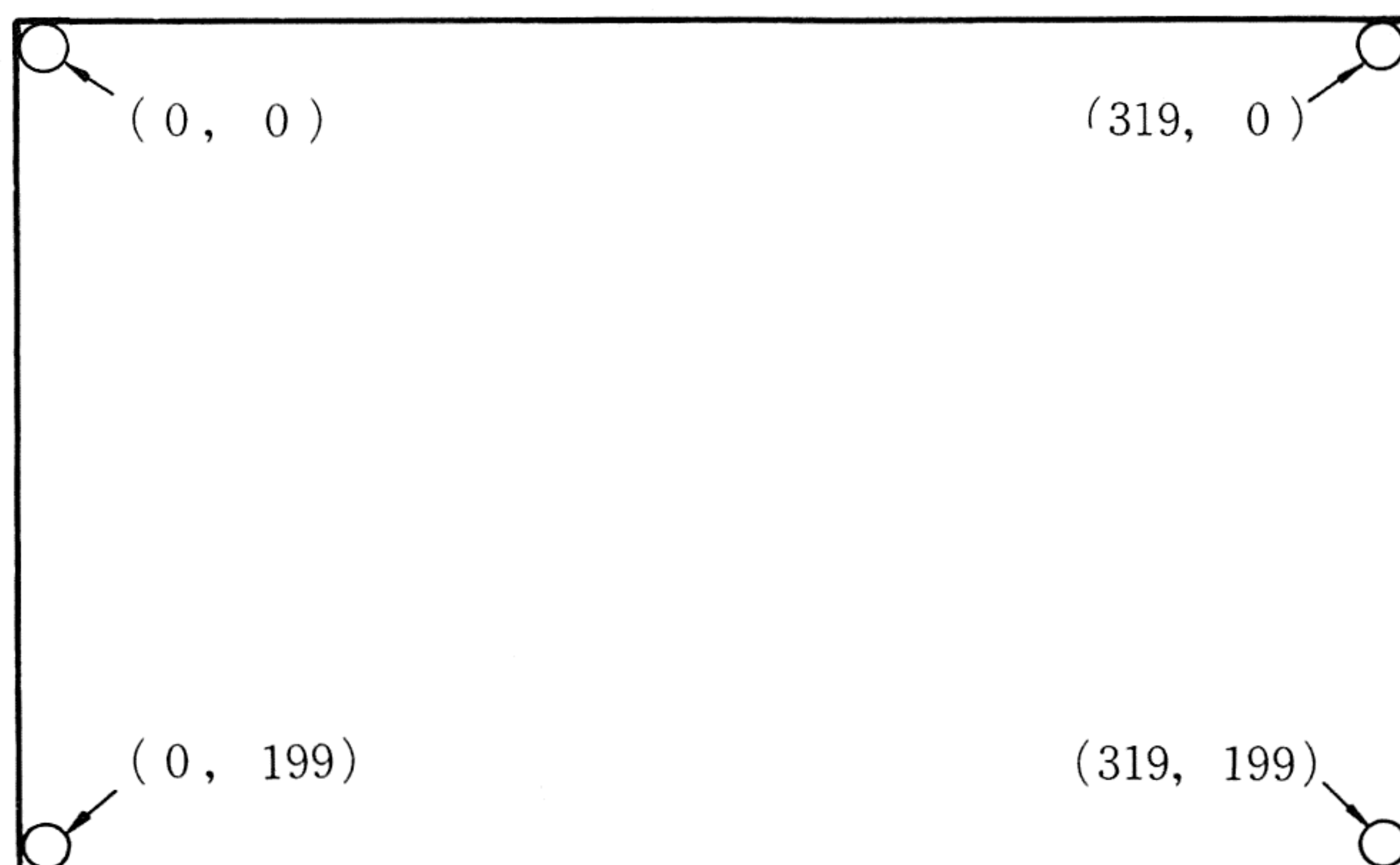
(0, 199)　(319, 199)

320×200ピクセル

図6.5　グラフィック座標

6.5　カラーコードとカラーナンバ

6.5.1　カラーコード(Color code)

MK II は，光の３原色（青，赤，緑）を組み合わせて８つの色を使うことができます。８色には，それぞれ**カラーコード**と呼ばれる色番号がつけられています。

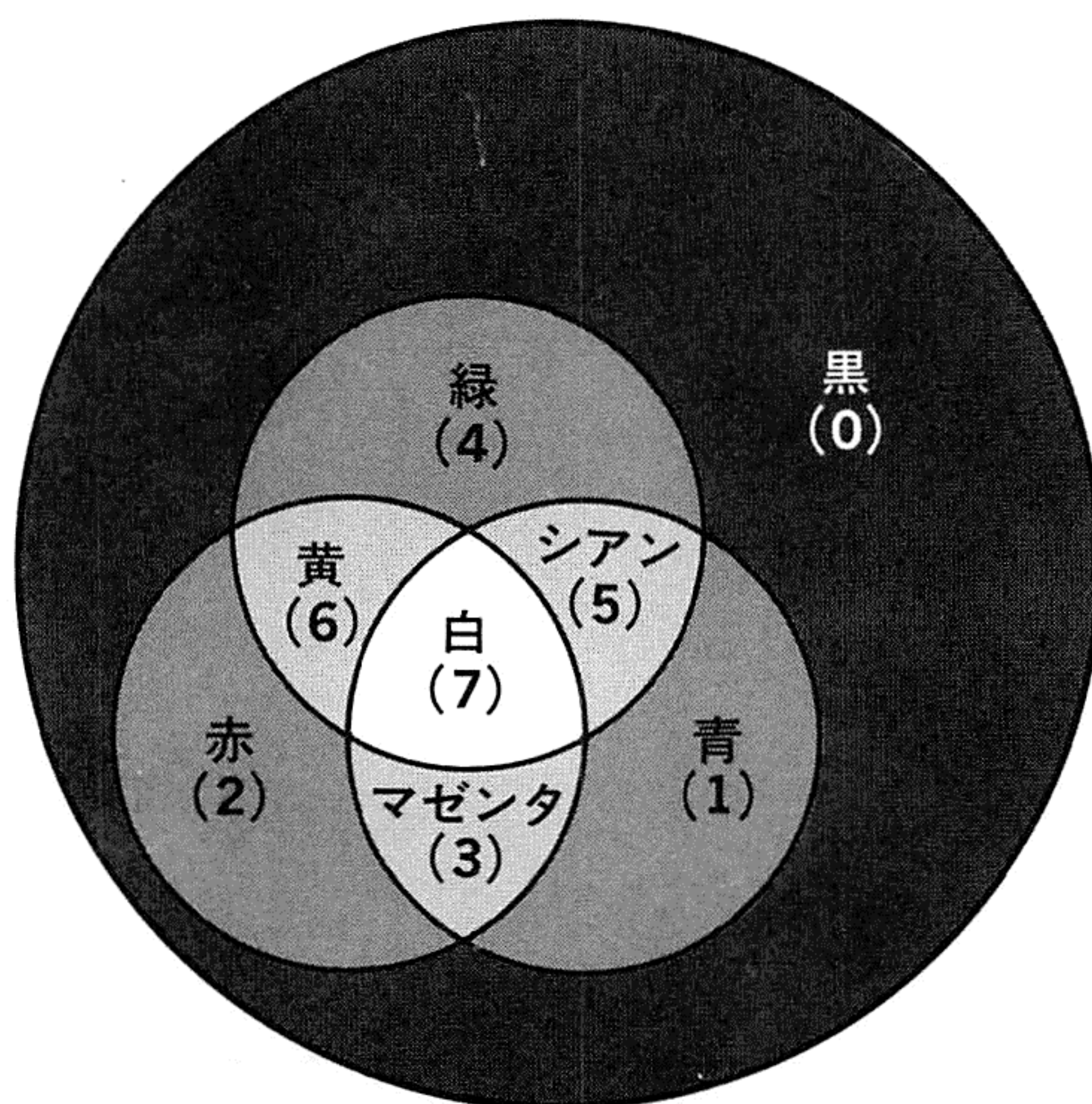



図６．６　８つの基本色とカラーコード

表６．１　カラーコード

３原色の組み合わせ	カラーコード	
	２進数	10進数
黒	０ ０ ０	０
青	０ ０ １	１
赤	０ １ ０	２
マゼンタ(＝赤＋青)	０ １ １	３
緑	１ ０ ０	４
シアン(＝緑＋青)	１ ０ １	５
黄（＝緑＋赤）	１ １ ０	６
白(＝緑＋赤＋青)	１ １ １	７

6.5.2 カラーナンバ(Color ♯)

カラーナンバも色につける番号ですが，カラーコードとは使い方が異なります。**カラーコードが絶対的な色番号であるのに対し，カラーナンバは，グラフィックモードによって対応する色が異なります。**6.8節で詳しく述べます。

6.5.3 カラーコードとカラーナンバの使い分け

表 6.2 に，カラーコードで色を指定する命令と，カラーナンバで色を指定する命令をあげます。

表6.2 カラーコードとカラーナンバの使い分け

カラーコードで色を指定する命令	
COLOR	テキスト画面に関する命令
LINE	
PSET	
PRESET	
CMD COLOR@	
CMD SCREEN	

カラーナンバで色を指定する命令
CMD CIRCLE
CMD COLOR
CMD LINE
CMD PAINT
CMD PRESET
CMD PSET
CMD PUT

6.6 フォアグラウンドカラーとバックグラウンドカラー

グラフィックスを描く，画用紙の色にあたるのが**バックグラウンドカラー**，えのぐの色にあたるのが**フォアグラウンドカラー**です。

MKII は，図 6.7 のアミの部分だけを，表示に使います。この部分いっぱいに，(正確にはビューポートに)，バックグラウンドカラーをぬりつぶし，その上に，フォアグラウンドカラーで，絵を描くわけです。

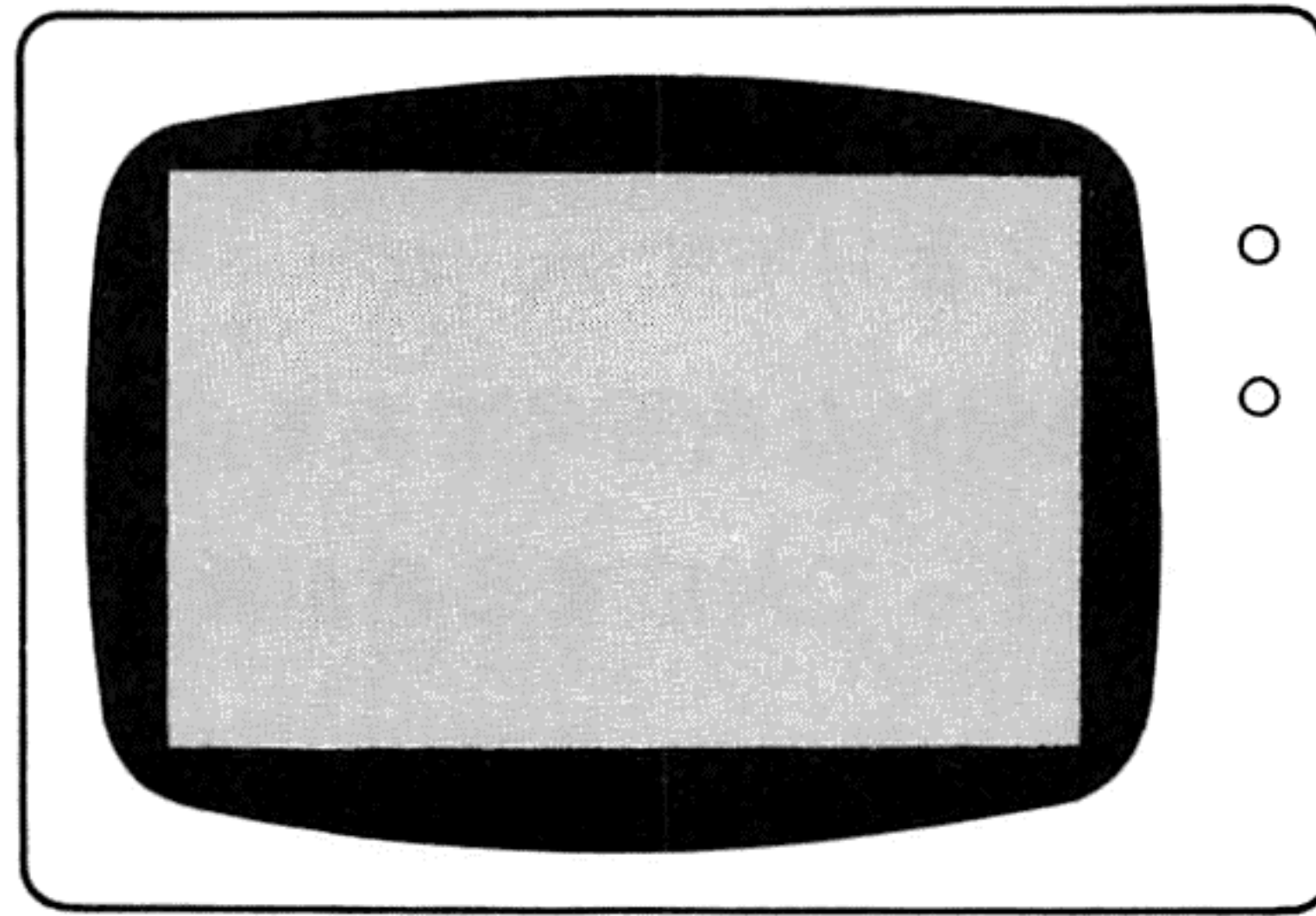



図6．7　ディスプレイ画面の表示領域

フォアグラウンドカラーとバックグラウンドカラーの設定は **CMD COLOR** で行います。

(書　式) **CMD COLOR 〈フォアグラウンドカラーナンバ〉, 〈バックグラウンドカラーナンバ〉**

(参　照) 6.9 節で具体的な使い方を述べます。

6.7　絶対座標とビュー座標

グラフィック画面は物理的に 640×200 ピクセル，または 320×200 ピクセルの表示ができますが，この中で実際に絵を描く長方形の範囲を**ビューポート**といいます。ビューポートにつけられる座標を**ビュー座標**といいます。ビュー座標に対して，図 6.5 に示した座標を**絶対座標**といいます。

通常，特に指定しなければ，絶対座標とビュー座標は一致しています。従ってこの場合ビューポートは640×200ピクセルまたは320×200ピクセルのグラフィックス画面全体になります。

ビューポートを設定する命令は **CMD VIEW** です。

(書　式) **CMD VIEW (〈左上の絶対座標〉) － (〈右下の絶対座標〉)**

絶対座標とビュー座標の関係は，次の例のようになります。

(Sample) 絶対座標の (0,100) から (249,199) までをビューポートに設定する。

CMD VIEW (0,100)－(249,199)

絶対座標(0, 0)

絶対座標(0, 100)
ビュー座標(0, 0)

ビューポート

絶対座標
(319, 199)

絶対座標(249, 199)
ビュー座標(249, 99)

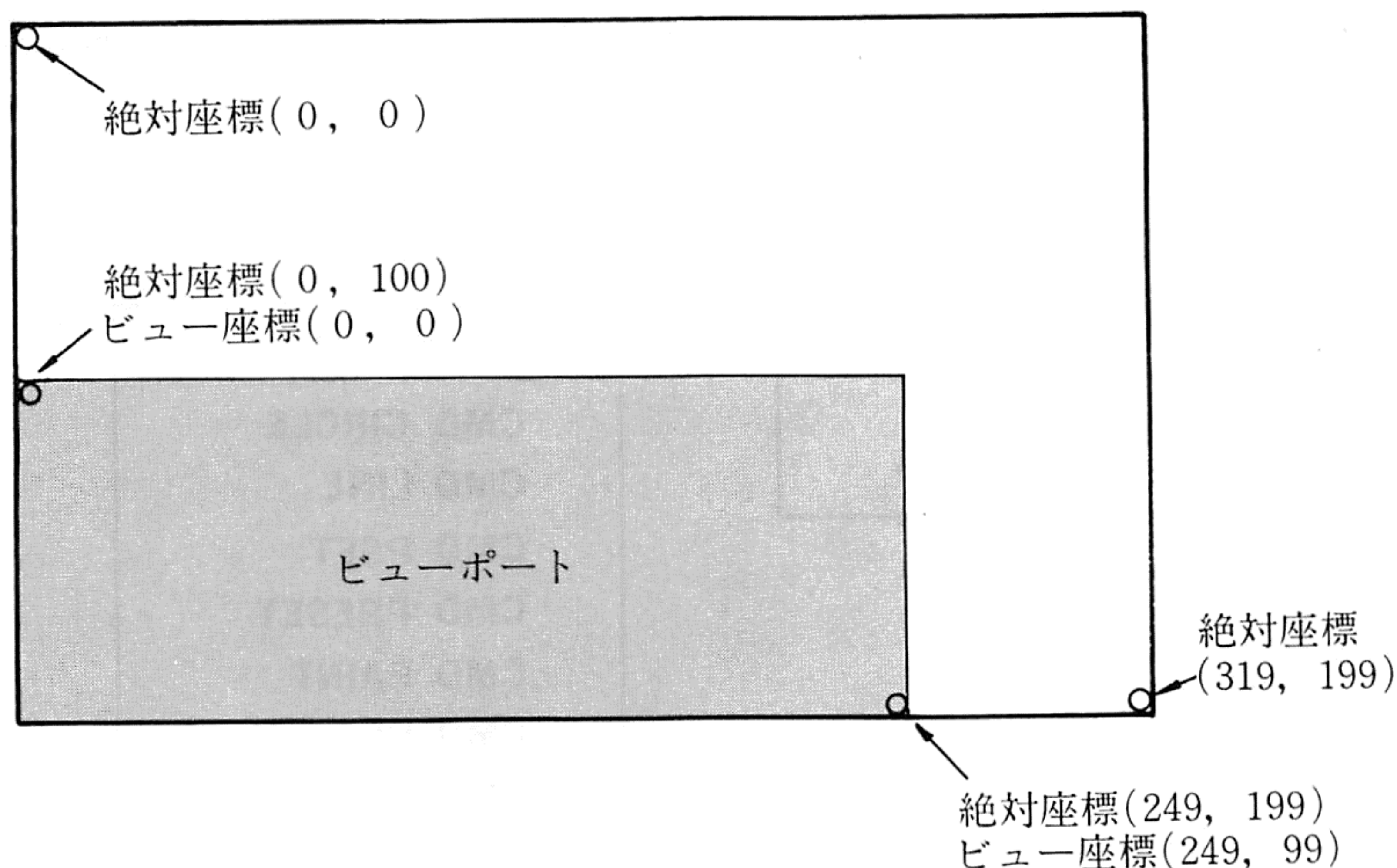



図6.8 ビューポートの設定

CMD LINE, CMD CIRCLE, CMD PSET などほとんどの高解像度グラフィックス命令は，このビュー座標の中で働きます(表6.3)。

Sample 上記の例のようにビューポートを設定したあと **CMD LINE** で (0,50) から (200,50) に直線をひくと，図 6.9 のようになります。

CMD LINE (0, 50)−(200, 50)

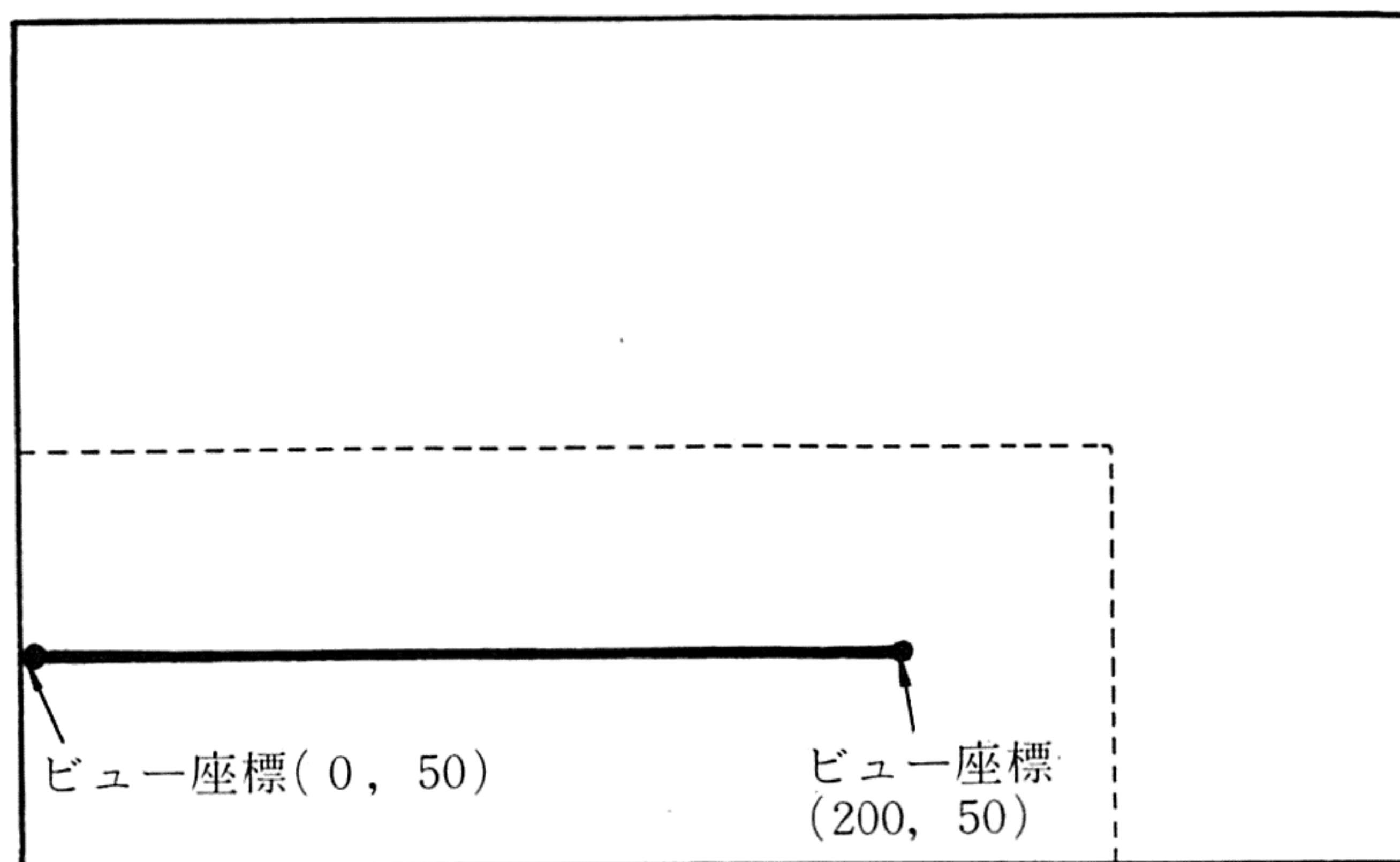
ビュー座標(0, 50)

ビュー座標
(200, 50)

図6.9 ビューポート内に直線をひく

グラフィックの命令や関数を絶対座標に関するものと，ビュー座標に関するものに分けて表 6.3 に示します。

表6.3　絶対座標とビュー座標

絶対座標に関するもの
CMD VIEW
STATUS VIEW

ビュー座標に関するもの
CMD CIRCLE
CMD LINE
CMD PSET
CMD PRESET
CMD PAINT
CMD PUT
CMD GET
CMD POINT
STATUS POINT

6.8　3つのグラフィックモード

ディスプレイ画面に，絵をかくときは次のようなグラフィック命令を使います。

CMD PSET	点を打つ。
CMD PRESET	点を消す。
CMD LINE	線や長方形をかく。
CMD CIRCLE	円，楕円，弧をかく。
CMD PAINT	色をぬる。
CMD PUT	登録しておいたパターンを呼び出す。

etc. ……

このようなコマンドを使って絵をかくときのモードは３つあり，それぞれのモードによって，コマンドの使い方が少しずつちがいます。この節では，この３つのグラフィックモードについて説明します。

３つのグラフィックモードの概要を，表 6.4 に示します。

表6.4　3つのグラフィックモード

グラフィックモード	モノクロモード	アトリビュートカラーモード	4色カラーモード	
			4色カラーモード0	4色カラーモード1
設定方法	**CMD SCREEN 0**	**CMD SCREEN 1**	**CMD SCREEN 2**	**CMD SCREEN 3**
解像度	640×200ピクセル		320×200ピクセル	
カラーナンバ (　)内は<選択色のカラーコード>を省略したとき選ばれる。	0：黒 1：選択色(白)	0：選択色(黒) 1：テキスト画面の色に依存する。	0：黒 1：赤 2：緑 3：選択色(青)	0：青 1：マゼンタ 2：シアン 3：選択色(黒)

絵をかく前に，まず，どのグラフィックモードを使うかを設定する必要があります。起動時にはモノクロモードに設定されています。グラフィックモードの設定には，**CMD SCREEN**を使います。

(書　式) **CMD SCREEN〔<グラフィックモード>〕〔,<グラフィックスイッチ>〕〔,<選択色のカラーコード>〕**

<グラフィックモード>　0～3の中から，グラフィックモードを選びます。

0……モノクロモード

1……アトリビュートカラーモード

2……4色カラーモード0

3……4色カラーモード1

モノクロモードは，細かい絵がかけますが，ひとつの画面で使える色は，黒と8色のうちの1色だけです。

アトリビュートカラーモードの解像度は，モノクロモードと同じです。カラーナンバ1はテキスト画面の色と同じになります。

4色カラーモードは，モノクロモードやアトリ

ビュートカラーモードほど細かい絵はかけませんが，ピクセル単位に色を指定できます。

〈グラフィックスイッチ〉 画面にグラフィックスを表示するかどうかを0，1のどちらかで指定します。

0……表示する。

1……表示しない。

このパラメータは，絵をかいている途中は表示させず，かき終わったら表示させる時などに切り換えて使います。

〈選択色のカラーコード〉 このパラメータで，選択色をカラーコード(0～7)で指定します。

起動時は，〈グラフィックモード〉 ＝0
〈グラフィックスイッチ〉＝0
〈選択色のカラーコード〉＝7

に設定されています。

CMD SCREENの３つのパラメータは，おのおの省略することができます。パラメータを省略したときは次のように設定されます。

① 〈グラフィックモード〉を省略したとき

(Sample) **CMD SCREEN ,1,1**

〈グラフィックモード〉は，それまでの状態と同じ値にセットされます。

② 〈グラフィックスイッチ〉を省略したとき

(Sample) **CMD SCREEN 2,,3**

〈グラフィックスイッチ〉は，それまでの状態と同じ値にセットされます。

③ 〈選択色のカラーコード〉を省略したとき

(Sample) **CMD SCREEN 3, 0**

〈グラフィックモード〉の値によって，必ず表 6.5 の値にセットされます。

表6.5　〈選択色のカラーコード〉の省略時の解釈値

〈グラフィックモード〉	〈選択色のカラーコード〉の省略時の解釈値
0	7(白)
1	0(黒)
2	1(青)
3	0(黒)

6.9　グラフィックス操作の基本手順

画面に絵をかくときの基本手順をフローチャートを使って説明します。

(はじめ)

グラフィックモードの設定。(CMD SCREEN)

3つのグラフィックモードのどれにするか，選択色を何色にするかを設定します。
それぞれのグラフィックモードで使われるカラーナンバに表6.4に示した色が対応づけられ，ビューポートは自動的に画面全体に設定されます。

ビューポートの設定。(CMD VIEW)

画面の一部分をビューポートに設定します。これにより絶対座標とは別にビュー座標が設定されます。**CMD VIEW** を実行しなければビューポートは画面全体に設定されたままです。

バックグラウンドカラー，フォアグラウンドカラーの設定。(CMD COLOR, CMDCLS 2)

CMD COLOR によってバックグラウンドカラーとフォアグラウンドカラーをカラーナンバで指定します。グラフィックモードの設定時に自動的に次のように設定されます。

グラフィックモード	フォアグラウンドカラー	バックグラウンドカラー
モノクロモード	カラーナンバ1	カラーナンバ0
アトリビュートカラーモード	カラーナンバ1	カラーナンバ0
4色カラーモード	カラーナンバ3	カラーナンバ0

絵をかく。
(CMD LINE,
CMD CIRCLE,
CMD PAINT,
CMD PSET,
CMD PUT, etc.)

CMD COLORで設定を変えたときは，**CMD CLS 2** を行なって バックグラウンドカラーを塗り変えてください。
ここで設定されたフォアグラウンドカラーは，次のステップで絵をかく際，色の指定を省略した場合に使われます。

ビューポート内に絵をかきます。位置はビュー座標で，色はカラーナンバで指定します。色の指定を省略すると，**CMD COLOR**で設定したフォアグラウンドカラーでかかれます。

絵を消す。
(CMD CLS 2)

おわり

図6．9　グラフィックス操作の基本手順

6.10 グラフィックス操作の例

これまで説明したように，**N80-BASIC** のグラフィックスは，少し複雑になっています。この節では，グラフィックモードごとに，実際に例をあげて使い方を説明します。

6.10.1 モノクロモード

表 6.4 で示したようにモノクロモードの解像度は 640×200 ピクセルで，使用できる色はカラーナンバ0（黒）とカラーナンバ1（8色のうち1色を選択する）の2色です。

Sample **10 CMD SCREEN 0, 0, 3**

- 3: カラーナンバ1をマゼンタ（カラーコード3）にする。
- 2番目の0: グラフィックス画面を表示する。
- 1番目の0: モノクロモードに設定する。

```
20 CMD COLOR 1, 0
```

- 1：フォアグラウンドカラーをマゼンタ（カラーナンバ1）にする。
- 0：バックグラウンドカラーを黒(カラーナンバ0)にする。

30 CMD CLS 2	グラフィックス画面をバックグラウンドカラーでぬりつぶす。
40 CMD CIRCLE(160, 100), 90	画面に円をかく。

（行番号 10） グラフィックモードをモノクロモードに，カラーナンバ1をマゼンタに設定しています。

（行番号 20, 30） カラーナンバで，バックグラウンドカラーとフォアグラウンドカラーを設定します。カラーナンバは0, 1以外の値を指定したときはその値を8で割ったあまりが0のときは0, それ以外のときは1とみなされます。設定後，画面をクリアします。

（行番号 40） フォアグラウンドカラーを省略していますから行番号 20で設定したフォアグラウンドカラーで円をかきます。フォアグラウンドカラーは，0以外の値はすべて1とみなされます。0を指定した場合は，バックグラウンドカラーで絵をかきますから，何も見えません。

6.10.2 アトリビュートカラーモード

表 6.4 で示したようにアトリビュートカラーモードの解像度はモノクロモードと同じ 640×200 ピクセルです。カラーナンバ1はテキスト画面の色と同じ（テキスト画面が白黒モードのとき，つまり **console,,,0** のときは白)になります。カラーナンバ0は8色のうちから1色を選びます。

図6.10 アトリビュートカラーモード

アトリビュートカラーモードでは，カラーナンバ1でかいた部分はテキスト画面の色になりますから，キャラクタ単位に色をつけ変えることができます。

(注　意) ただし，一行の中で 20 回以上色を変えることはできません。

アトリビュートカラーモードのグラフィックスにキャラクタ単位に色をつけるには **CMD COLOR@** を使います。この場合は，**CONSOLE** でカラーモードにしておく必要があります。以下の例を参考にしてください。

(Sample)

10 CONSOLE 0, 25, 0, 1	テキスト画面をカラーモードにする。
20 WIDTH 40, 25	テキスト画面にキャラクタ座標を設定する。
30 CMD SCREEN 1, 0, 1	
最後の 1	カラーナンバ0を青（カラーコード1）にする。
最初の 1	アトリビュートカラーモードに設定する。
40 CMD COLOR 1, 0	
0	バックグラウンドカラーを青（カラーナンバ0）にする。
1	フォアグラウンドカラーをテキスト画面の色（カラーナンバ1）にする。
50 CMD CLS 2	グラフィック画面をバックグラウンドカラーでぬりつぶす。
60 CMD COLOR@ (0, 0)−(19, 9), 2	キャラクタ座標の（0，0）から（19，9）までを赤（カラーコード2）に設定する。
70 CMD COLOR@ (20, 10)−(37, 24), 3	キャラクタ座標の（20，10）から（37，24）までをマゼンタ（カラーコード3）に設定する。
80 FOR I=0 TO 3000 STEP 13 **90 CMD LINE (1, 0)−(I, 199)** **100 NEXT I**	フォアグラウンドカラーで直線をひく。

（行番号 10 ） **CONSOLE** の４番目のパラメータを１にすると，テキスト画面はカラーモードになります。テキスト画面が白黒モードの場合は，**CMD COLOR @** を使って色の指定をすることはできません。

（行番号 20 ） **WIDTH** でキャラクタ座標を設定します。この例では，40×25 文字にしています。このキャラクタ座標上に，**CMD COLOR @** で色をつけるのです。

（行番号 30 ） グラフィックモードをアトリビュートカラーモードに，カラーナンバ０を青に設定しています。

（行番号 40, 50） カラーナンバでバックグラウンドカラーとフォアグラウンドカラーを設定します。０，１以外の値を設定した場合はその値を８で割ったあまりが０のときは０，それ以外のときは1とみなされます。設定後，画面をクリアします。

（行番号 60, 70） キャラクタ座標上の長方形の部分に，カラーコードで色をつけます。

（行番号 80, 90, 100 ） フォアグラウンドカラーを省略していますから，行番号 40 で設定したフォアグラウンドカラーで直線をひきます。 フォアグラウンドカラーは，０以外の値はすべて１とみなされます。０を指定すると，バックグラウンドカラーで絵をかきますから何も見えません。

注　意 この例でおわかりのように，**CMD COLOR @** は，グラフィック画面の端から端まで色をつけることは，実はできないのです。 画面の両端には色がつけられません。 たとえば **WIDTH** で 40×25 文字に設定した場合には，左上の(０,０)から右下の(37,24)の範囲内だけで，色指定を行います。

参　照 **N80-BASIC** リファレンスマニュアル

グラフィック画面

(0, 0)

両端に色がつけられない

(37, 24)

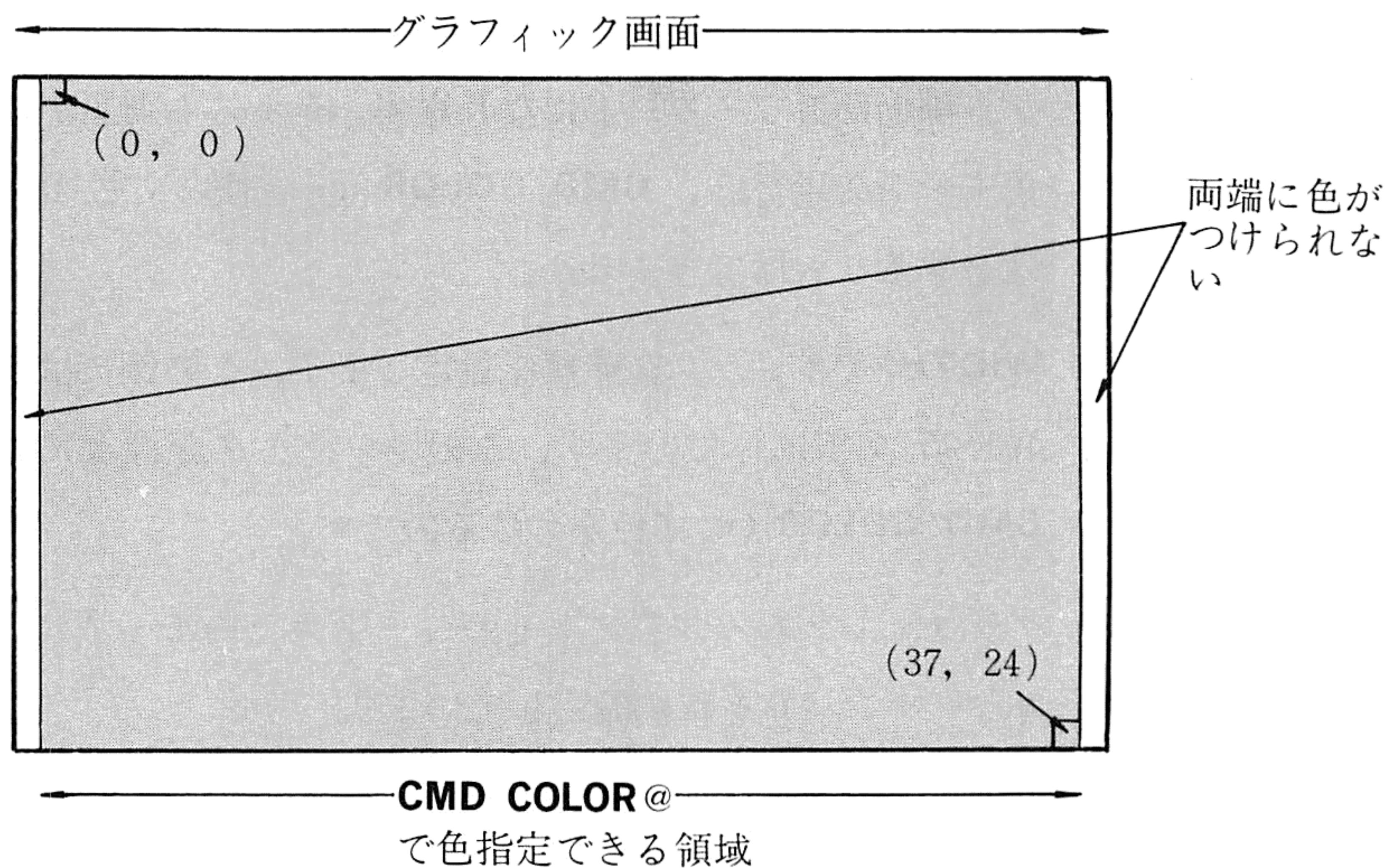
CMD COLOR@で色指定できる領域

(WIDTH 40, 25と設定したとき)

図6.11 **CMD COLOR**@で色指定できる領域

6.10.3 4色カラーモード

4色カラーモードの解像度は 320×200 ピクセルです。

4色カラーモードには，4色カラーモード0と4色カラーモード1との2つのモードがあります。それぞれカラーナンバは0から3まであります。カラーナンバ0，1，2にはあらかじめ対応する色が決められており，カラーナンバ3はユーザが指定します。

表6.6 4色カラーモードのカラーナンバ

グラフィックモード / カラーナンバ	4色カラーモード0 **CMD SCREEN 2**	4色カラーモード1 **CMD SCREEN 3**
0	黒	青
1	赤	マゼンタ
2	緑	シアン
3	選択色(青)	選択色(黒)

```
(Sample) 10 CMD SCREEN 2, 0, 6
                      ↑    ↑
                      │    └カラーナンバ3を黄（カラーコード
                      │     6）にする。
                      └────4色カラーモード0に設定する。
         20 CMD COLOR 3, 0
                      ↑  ↑
                      │  └────バックグラウンドカラーを黒(カラー
                      │        ナンバ0)にする。
                      └───────フォアグラウンドカラーを黄(カラー
                               ナンバ3)にする。
         30 CMD CLS2           グラフィック画面をバックグラウンド
                               カラーでぬりつぶす。
         40 CMD CIRCLE (60, 100), 40, 2      ┐
         50 CMD CIRCLE (160, 100), 40        │
         60 CMD CIRCLE (260, 100), 40, 1     │
         70 CMD PAINT  (60, 100), 2          ├ 絵をかく。
         80 CMD PAINT  (160, 100)            │
         90 CMD PAINT  (260, 100), 1         ┘
```

（行番号 10 ）　グラフィックモードを4色カラーモード0に，カラーナンバ3を黄（カラーコード6）に設定しています。

（行番号 20 ）　バックグラウンドカラーとフォアグラウンドカラーの設定をしています。0，1，2，3以外の数を指定すると，4で割ったあまりとして設定されます。

6.11　低解像度グラフィックス(N-BASICのグラフィックス)

この章のはじめに触れたように，**N-BASIC**ではグラフィック専用のVRAMを使わず，グラフィックスは，テキストのVRAMに書き込まれます。このグラフィックスを低解像度グラフィックスと呼び，**N80-BASIC**，**N-BASIC**の両方とも，この機能をもちます。低解像度グラフィックスに使う命令や関数を次に列挙します。

WIDTH
CONSOLE
COLOR
LINE
PSET

PRESET
PUT@
GET@
POINT
etc.

6.11.1 低解像度グラフィック座標

低解像度グラフィックスのひとつのピクセルは，1つのキャラクタを表示するスペースを図6.12のように8等分したものです。

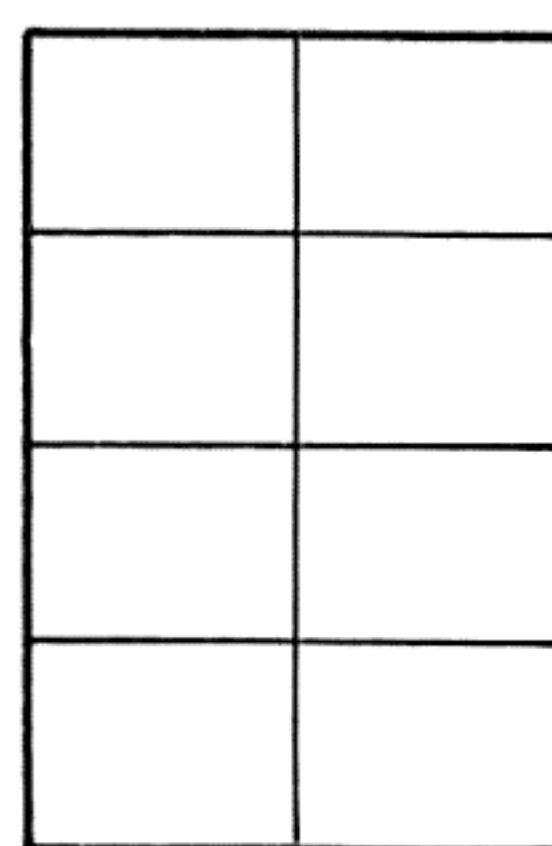



図6.12

したがって，解像度は，**WIDTH** で設定したキャラクタ座標に依存します。

WIDTH 36, 25 72×100ピクセル
WIDTH 40, 25 80×100ピクセル
WIDTH 72, 25 144×100ピクセル
WIDTH 80, 25 160×100ピクセル

(**注　意**) **WIDTH** で20行／画面に設定すると，行間に1ピクセル分のすきまができてしまうので注意してください。

6.11.2 モードの設定

モードの設定は，次の①～③について行う必要があります。①～③は，特にこの順に行う必要はありません。

① **CONSOLE** でカラーモードにするか白黒モードにするかを設定します。

Sample CONSOLE ,,,1　カラーモード
CONSOLE ,,,0　白黒モード

② **WIDTH** で解像度を設定します。

③ **COLOR** でグラフィックモードに設定します。

Sample COLOR ,,1
PRINT CHR$(12);

①でカラーモードにした場合と白黒モードにした場合とでは，**COLOR** の第1パラメータで指定する機能が異なります。

③でグラフィックモードに設定したあとには，**PRINT CHRS$(12);**を実行する必要があります。 ダイレクトモードでは， HOME CLR キーを押します。グラフィックモードに設定しなくてもグラフィック命令の実行ができない訳ではないのですが，グラフィックモードにしておくと動作が保障されます。

6.11.3 諸注意

(1) **白黒モード**

キャラクタだけ，もしくはグラフィックス(4．2．1節のグラフィックシンボルではありません)だけを表示する場合には問題ありません。

キャラクタとグラフィックスを混在で表示を行う場合，グラフィックスの1ピクセルは，1つのキャラクタを表示するスペースを 図6.13 のように8等分にしています。従ってキャラクタを表示しているポジションの所にグラフィックスを表示しますと， 後から表示した方が優先になりますのでポジション に注意してください。

これは，カラーモードでも同じです。

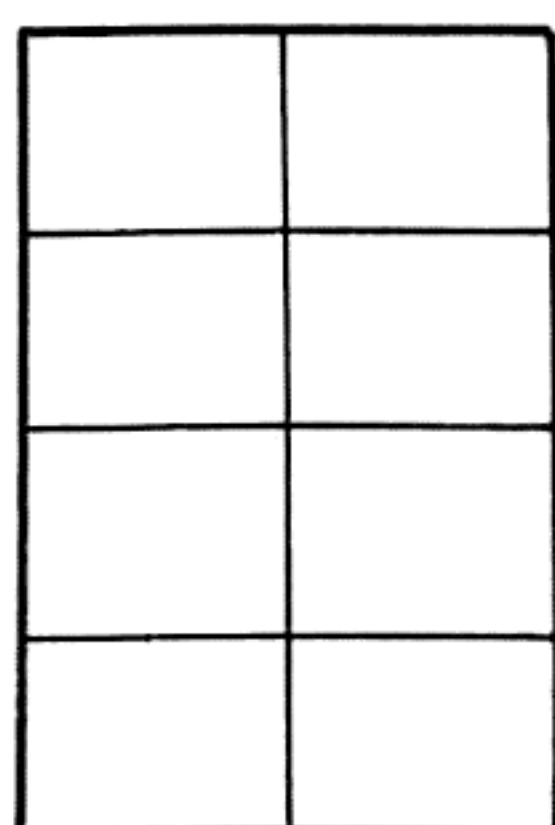



図6．13

また，**COLOR**の<グラフィックスイッチ>を，キャラクタモードにしている場合同一行で文字もしくはスペース（ブランク）とグラフィックス表示の変化点の指定が20ヶ所を超えるとそれ以後（同一行）グラフィックスが表示されなくて文字が表示されてしまいますので注意してください。グラフィックモードにしている場合は，グラフィックスもしくはスペース（ブランク）と文字との変化点の指定が20ヶ所を超えるとそれ以後(同一行)は，グラフィックスが表示されてしまいます。

画面にグラフィックスをキャラクタ（文字）より多く表示する場合は，グラフィックスイッチをセットして画面のクリアを行っておいた方が上のような問題が起こりにくくなります。

(2) カラーモード

キャラクタ表示だけ(同色)もしくは，グラフィックス表示だけ(同色)の場合は問題ありませんが，キャラクタ表示とグラフィックス表示の混在及び色を変化させる場合に注意が必要です。

キャラクタとグラフィックスを混在で表示する場合，表示するポジションに注意してください(白黒モードと同じ)。

グラフィックスに色をつける場合は,ひとつのキャラクタを表示する領域に存在する8ピクセルは同色になります。従って,1つのピクセルの色を指定すると,隣接する7つのピクセルの色が同時に変化してしまいます。

また，同一行において色の変化点が19ヶ所を超えた場合，それ以後(同一行)は同一の色になってしまいます。

したがって，多く使う色を**COLOR**コマンドでセットして画面をクリアしておいた方がいいと思います。

次に，キャラクタ(文字)とグラフィックスを混在させる場合は，白黒モードと同様ですが，変化点の数が19ヶ所になります。

以上の点に注意してプログラムを作ってください。（プログラムの実行で画面をクリアする場合は，プリント文で行います。)

```
PRINT CHR$(12) ;
```

第7章
カセットテープの使い方 (ROM-BASICモード)

この章では，カセットテープを使って，データやプログラムを保存する方法，逆にカセットテープに書き込まれたデータやプログラムを **MK II** に読み込む方法について説明します。

カセットテープは，BASIC のコマンドによって使う場合と，**10 章 モニタ** の中で説明するように機械語のプログラムの読み書きに使う場合が考えられます。この章では，BASIC でカセットテープを使う場合だけについて説明します。

7.1 ROM-BASICモード

BASIC モードには，**ROM-BASIC モード**と，**DISK-BASIC モード**との2つがあるということは，**第5章 ソフトウエア概説** で述べました。カセットテープは，どの BASIC モードでも全く同じように使うことができますが，一般的に，DISK-BASIC モードでは，あまり使われません。なぜなら，DISK-BASIC モードでは，わざわざカセットテープを使うより，フロッピィディスクを使う方が便利だからです。この節では，カセットテープが不可欠である ROM-BASIC モードの概要について説明します(DISK-BASIC モードについては8章で説明します)。

ROM-BASIC モードを起動するには，3.3 電源の投入と起動テストで述べたように，フロッピィディスクユニットを接続しないで行います。

MK IIのマイクロプロセッサ **μPD-780C-1** は，64K バイトのメモリ空間を直接読み書きすることができます。BASIC が動いているときは，

64K バイトのうち下位（アドレスの若い方）の半分は，**ROM**（Read Only Memory）と呼ばれるメモリが選択されます。ROM は，名前のとおり読むことしかできないメモリです。ユーザは ROM にデータを書き込むことはできません。ROM には，BASIC が動くためのプログラム（BASIC インタプリタといいます）が書き込まれていて，電源を切ってもプログラムは消えません。

ROM に対して，64K バイトのうち上位の半分を構成するメモリを **RAM**（Random Access Memory）といいます。RAM は，読むことも書くこともできるメモリーで，ユーザの作ったプログラムはここに書き込まれます。電源を切ったり，リセットボタンを押したりすると，RAMの内容は消えてしまいます。

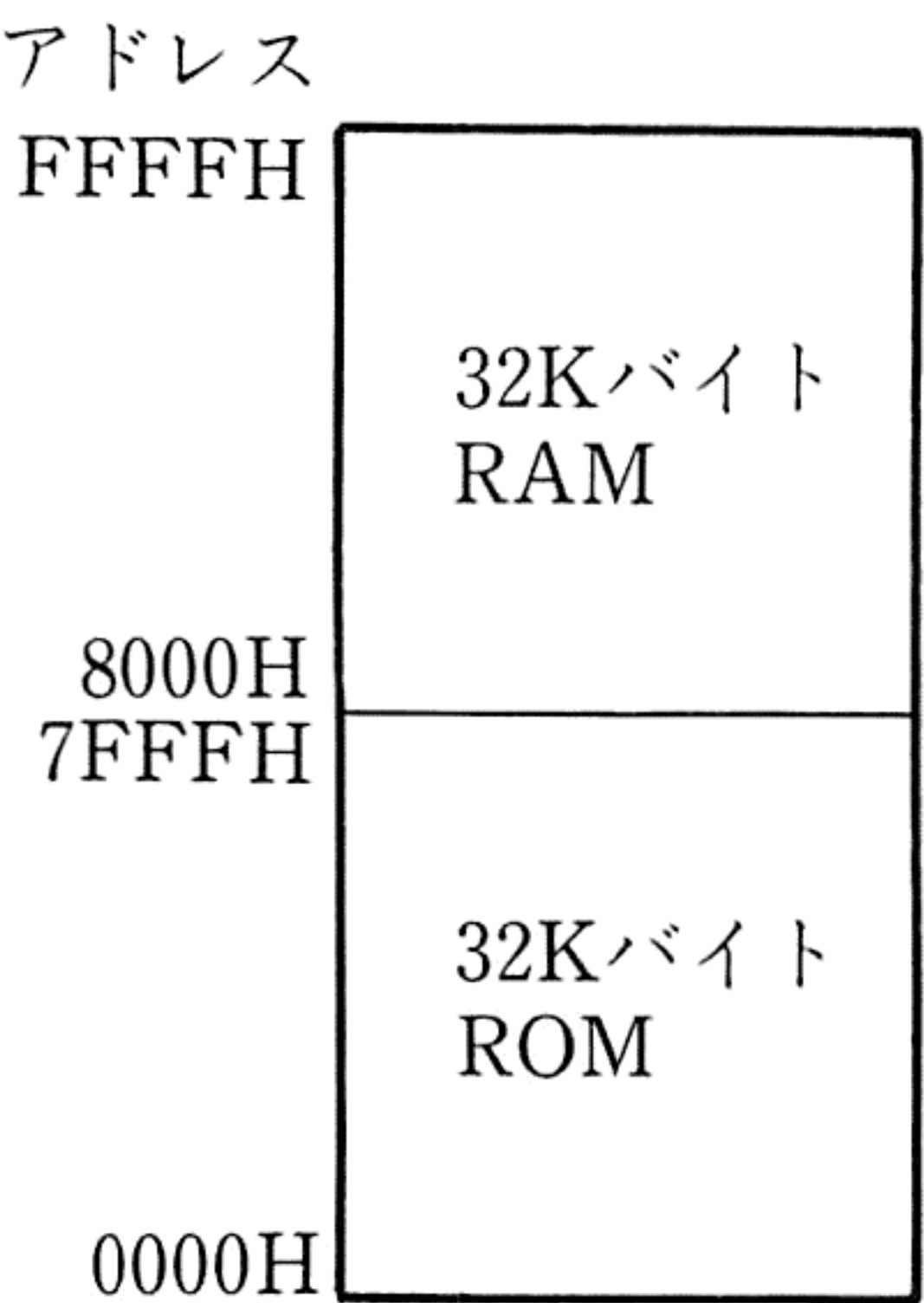



図７．１　ROMとRAM

前置きが長くなりましたが，ROM-BASIC モードとは，ROM の 32K バイトに書き込まれている BASICインタプリタが作動しているモードです。

それに対して，ROM の 32K バイトに，フロッピィディスクから，RAM の一部に読み込んだものを加えて，大きくなった BASIC インタプリタが作動するモードが，DISK-BASIC モードです。

ユーザの入力のしたプログラムやデータは，RAM に書き込まれます。しかし，**MKII**の電源を切ったり，リセットボタンを押したりすると，RAM の内容は初期化されてプログラムなどは消えてしまいます。したが

って，再び使用する可能性のあるプログラムは，フロッピィディスクやカセットテープに保存しておきます。そうすれば，電源を切ったり，リセットボタンを押したりするたびに最初からプログラムやデータをキー入力する必要はなくなります。メモリから，フロッピィディスクやカセットテープにプログラムやデータを書き込むことを“**セーブする**”といいます。逆に，フロッピィディスクやカセットテープからメモリに読み込むことを，“**ロードする**”といいます。

Note
セーブ　：　メモリ　→　外部の記憶装置
ロード　：　外部の記憶装置　→　メモリ

7.2　カセットテープの使い方

カセットテープは，**MKII** のすべての BASIC モードで全く同じように使えます。使用するテープレコーダは，一般用オーディオカセットテープレコーダです。テープレコーダとしては，パーソナルコンピュータ用に設計されたデータレコーダ（**PC-6081，PC-6082**）が操作しやすく，失敗も少ないようですが，家庭用のものでも十分機能が果たせます。

7.2.1　テープレコーダの接続

3．2 周辺装置との接続に従って **MKII** とカセットテープレコーダを接続します。このとき，次の点に注意してください。

(1)　リモート端子（黒色）を接続した場合。

リモート端子を接続しますと，テープレコーダのモータの **on/off** は，**MKII** がコントロールします。ですから，［PLAY］ボタンを押していても，カセットテープを使用する命令が実行されなければ，カセットテープは回りません。この機能によって，必要な間だけ，カセットテープを回すことができ，カセットテープの無駄使いが防げ，また，手間もかかりません。

(2) リモート端子を接続しなかった場合

この場合は，テープレコーダのスタート，ストップは，すべてユーザが操作します。ですから，カセットテープを使用する命令を実行する前に PLAY ボタン(と REC ボタン)を押し，終ると STOP ボタンを押すという操作が必要になります。

普通，1つのプログラムをロードしたり，セーブする場合は，リモート端子を接続せずに操作することが多いようですが，プログラムの中でデータのロード，セーブを行う場合は，リモート端子を接続していないと，うまく動作しない場合があります。以後，リモート端子が接続されているとして説明をします。

7.2.2 プログラムのセーブ

MK II とテープレコーダが接続できたら，キーボードから，次のプログラムを入力してください(入力が終っても実行させないでください)。

```
new RETURN
10  INPUT A, B, C, D RETURN
20  PRINT "Set tape, then push PLAY and REC" RETURN
30  INPUT "ok" ; A$ RETURN
40  PRINT #−1, A, B RETURN
50  PRINT #−1, C, D RETURN
60  END RETURN
```

入力したプログラムがまちがっていないかどうか，ディスプレイとよく見比べてください。まちがっていなければ，このプログラムをカセットテープへセーブしましょう。

(注　意) カセットテープにセーブするプログラムは最後に **END** をつける習慣をつけてください。 **END** がついていないとロードしたときに実行できないことがあります。

1．カセットテープをテープレコーダにセットします。

プログラムやデータを入力すると，当然，前に録音してあったものは消えてしまいます。また，このとき，テープカウンタを0にセットし

ておけば，どこからセーブしたのかすぐにわかります。

2．テープがセットできたら [REC] ボタンと [PLAY] ボタンを押します。リモート端子（黒）が接続してあれば，カセットテープは回り始めないはずです。そこで次の命令を入力します。

motor [RETURN]

[RETURN] キーを入力すると，“カチッ”という音がして同時にテープが回り始めます。数秒たったらもう一度，

motor [RETURN]

を入力します。すると再び“カチッ”という音がして，テープは止ります。これでセーブの準備が整ったわけです。

（注 意）カセットテープには，始めと終りにクリアフィーダという磁性体が塗ってない部分があります。この部分には記録できませんから，プログラムやデータをセーブする前にスキップしておく必要があります。また，前のプログラムやデータとの区別をつけやすくするためにも，セーブする前は，**MOTOR** でテープを送る習慣をつけましょう。

Note **MOTOR は，モーターをコントロールします**

MOTOR [RETURN] ⇨モータースタート
⇨ MOTOR [RETURN] ⇨モーターストップ

3．いよいよプログラムをセーブします。[PLAY] ボタンと [REC] ボタンは押したままになっています。そこで，次のようにキーボードから入力します。

csave"test" [RETURN]

4．しばらくすると“**Ok**”が表示され，プログラムのセーブは終りました。

ここで使用した **CSAVE** がプログラムを「セーブ」する命令です。

（書 式）**CSAVE"<ファイル名>"**

今の場合ですと **test** がファイル名です。

カセットテープへのプログラムのセーブは CSAVE で行います。

CSAVE"XXXXXX"

XXXXXX ： 6文字以内のファイル名

7.2.3 ベリファイ

カセットテープにプログラムをセーブした場合は，正しくセーブされたかどうかチェックする必要があります。つまり，メモリの中のプログラムとカセットテープにセーブされたプログラムが等しいかどうか比べるわけです。これを**ベリファイ**するといいます。ではベリファイの実行のしかたを順を追って説明しましょう。

(注　意) ベリファイはプログラムをセーブした後，すぐに行ってください。セーブした後，**RUN** してからベリファイをすると，メモリの内容が変わるためエラーになることがあります。

1．プログラムのセーブを始めたところまでテープを巻き戻します。リモート端子を接続している場合は **MOTOR** を使います。巻き戻しが終わったら，モータストップ（**MOTOR** [**RETURN**]）を実行しておきます。

2．テープレコーダの音量と音質を調整します。音量は，普段使用する場合よりもやや大きめにセットし，音質が調整できる場合は，高音が強調される側へセットします。

3．[PLAY] ボタンを押した後で，次の命令を実行します。

cload ? "test" [**RETURN**]

4．"**test**" というプログラムが見つかると

Found : test

と表示し，読み込んだデータとメモリの内容をチェックします。

5．カセットテープにセーブされている内容が一致した場合は，

Ok

と表示し，一致しない場合は，

Bad

と表示してきます。

一致しなかった原因としては，次の場合が考えられます。

1．ベリファイの方法がまちがっていた。

テープの巻き戻しを忘れていたり，セーブしてすぐにベリファイをしなかったときなどです。

2．再生のときの音量，音質の調整が適当でなかった。

この場合は，音量と音質を調整し直して，何度か挑戦してみてください。使用しているテープレコーダによっては，音量が，ある特定の範囲でしか，うまく動作しない可能性があります。

3．セーブがうまくできていなかった。

音量，音質をいくら調整しても **Bad** と表示される場合は，セーブした時に，失敗した可能性があります。その原因としては，次のような事が考えられます。

(1) ケーブルの接続が正しくなかったり，接続プラグが，しっかりと差し込まれていなかった。

(2) 録音音量が調整できるテープレコーダを使用したが，その調整が正しくなかった。

(3) セーブするときに，カセットテープのクリアフィーダの部分をスキップするのを忘れていた。

それぞれ，もう一度よく確認してから，セーブを行ってください。これらの原因を検討してもセーブできない理由がわからない場合は，お近くの**Bit-INN**もしくは，お買い求めの販売店で御相談ください。

Note **プログラムをカセットテープにセーブした場合は，その直後に必ずベリファイを実行して，確認しましょう。**

7.2.4 プログラムのロード

今度は，カセットにセーブしたプログラムを，再びメモリへロードしてみましょう。今はまだ，メモリにプログラムが残っていますから，

[f･4] [RETURN]（または，list [RETURN]）

を入力すれば，メモリに入っているプログラムが表示されます。そこでまず最初に，メモリに入っているプログラムを消します。

new **RETURN**

この命令を実行すると，あなたが入力したプログラムは，すべて消えてしまいます。ですから，

f・4 **RETURN**（または **list** **RETURN**）

を入力しても，プログラムは表示されません。では，ロードを始めます。

1．カセットテープを巻き戻します。（リモート端子を接続している場合は，**MOTOR** **RETURN** を入力しないとモータは回りません）

2．音量と音質を確認します。（ベリファイしたときと同じです）

3．次の命令をキー入力して，**RETURN** キーの入力と同時にテープレコーダの PLAY ボタンを押します。

cload "test" RETURN

4．するとカセットテープが回り始めます。プログラムが見つかると

Found : test

とディスプレイに表示し，ロードしている間は，画面の右上スミに“*”を表示します。指定した以外のプログラム名が見つかると，

Skip : XXXXXX

と，プログラム名を表示して，次のプログラムを読みに行きます。

5．ロードが無事に終了すれば，

Ok

が表示され，カセットテープは止まります。

もし，ロードが途中で失敗すれば，ディスプレイに，

Tape read ERROR

というメッセージが表示され，ブザーが鳴ります。

エラーとなった場合には，ベリファイのところで説明した原因を１つ１つチェックしてみましょう。何度，ロードしてもエラーとなる場合は，テープにキズがある場合などが考えられます。また，ベリファイを行っていなければ，うまくセーブができていない可能性があります。このような場

合は，もう一度，セーブからやり直す必要があります。

ロードがうまく行けば，

f・4 **RETURN**（または **list** **RETURN**）

でプログラムが表示されます。

注　意 ロードがうまくいかなかった場合に，

f・4 **RETURN**（または **list** **RETURN**）

を実行すると，途中までのプログラムが表示される場合がありますが，このプログラムを実行させたり，新たな行をつけ加えて実行させることはできません。 **NEW** を実行して，最初から入力してください。

7.2.5 データのセーブ

1．プログラムのセーブとロードに用いたプログラムをロードします。

2．このプログラムを実行させます。

f・5（または，**run** **RETURN**）

3．すると画面に，“？”を表示します。これは，行番号 10 の **INPUT** が，キーボードからの入力を要求しているのです。（**INPUT** については，「N80-BASIC リファレンスマニュアル」を参照してください。）そこで，

1, 2, 3, 4 **RETURN**

と入力します。

4．すると今度は，次のメッセージが表示されます。

```
Set tape, then push PLAY and REC
ok ?
```

5．そこで，カセットテープをテープレコーダにセットします。このとき，テープカウンタを0にしておけば，後でどこからデータが入っているか一目でわかり，とても便利です。

6．カセットテープのセットが終ったら，テープレコーダの REC ボタンと PLAY ボタンを押します。これらの準備が終ったら，

RETURN キーを入力します。

7．RETURN キーを入力すると，すぐに，カセットテープが回りはじめ，しばらくして，ディスプレイに“**Ok**”と出力し，カセットテープは止まります。

これで，キーボードから入力した数値が，カセットテープにセーブされたわけです。

では，このプログラムを簡単に説明しましょう。

```
10 INPUT A, B, C, D
20 PRINT "Set tape, then push PLAY and REC"
30 INPUT "ok" ; A$
40 PRINT #-1, A, B
50 PRINT #-1, C, D
60 END
```

キーボードから数値を入力し，カセットテープのセットも終り，REC ボタンと PLAY ボタンを押して，RETURN キーを入力します。(行番号 10，20，30)

PRINT#-1 でデータをセーブします。(行番号40，50)

PRINT は，ディスプレイに文字を表示させるだけではなく，このように #-1 をつけることでカセットテープへのデータの出力を行います。

7.2.6 データのロード

7.2.5でセーブしたデータをロードしてみましょう。

1．次のプログラムを入力します。

```
new RETURN
10 A=0 : B=0 : C=0 : D=0 RETURN
20 PRINT "push PLAY" RETURN
30 INPUT #-1, A, B RETURN
40 INPUT #-1, C, D RETURN
50 PRINT A, B, C, D RETURN
60 END RETURN
```

このプログラムとデータのセーブのプログラムとの違いは，行番号

30 と 40 でデータのロードを "**INPUT ♯－1**" で行っている点です。

2．プログラムが入力できたら，データをセーブし始めたところまで，カセットテープを巻き戻します。

3．カセットテープの準備ができたら，実行させます。

　　[f·5]（または，　**run** [RETURN]）

4．"**push PLAY**" というメッセージが現われたら，[PLAY] ボタンを押します。カセットテープが回りはじめて，しばらくすると，ロードされたデータがディスプレイに表示されます。

7.2.7　データの型と数の一致

カセットテープにデータをセーブ，ロードする場合の注意事項として，セーブするときとロードするときの〈データの型と数の一致〉があります。データとして文字をセーブしたのに，それを数値データとしてロードすることはできません。また，1つの **PRINT** で4つのデータをセーブしているのに，**INPUT** では2つしかロードしないというわけにもいきません。ただし，データの型と数が一致していれば，ロードしてきたデータを代入する変数は，セーブしたときに使った変数と異なっていてもかまいません。

(**Sample**)（良い例1）

セーブ

```
100 PRINT  ♯－1, A, B, C
```

ロード

```
150 INPUT  ♯－1, X, Y, Z
```

データの型（数値），数（3つずつ）が一致しているので OK です。

(**Sample**)（良い例2）

セーブ

```
100 PRINT  ♯－1, YAMA$, KAWA$
```

ロード

```
200 INPUT  ♯－1, SORA$, UMI$
```

これも，データの型(文字)，数(2つずつ)が一致しているのでOKです。

Sample（悪い例）

セーブ

```
100 PRINT #−1, a, b, c, d
```

ロード

```
200 INPUT #−1, a, b
210 INPUT #−1, c, d
```

データの型（数値）は一致していますが，1つの **PRINT** で4つのデータをセーブしたのに対し，ロードするときは，1度に2個しか扱っていません。これでは，ロードできません。

Note **データのセーブとロードには，データの型と数の一致が必要です。**

第8章
フロッピィディスクの使い方（DISK-BASICモード）

この章では，フロッピィディスクユニットを接続したシステムで起動する DISK-BASIC モードについて説明します。

8.1　DISK-BASICモードの概要

DISK-BASIC モードで使用するには，フロッピィディスクユニットとシステムディスクが必要です。（**N80 DISK-BASIC** と **DISK-BASIC** をあわせて DISK-BASIC モードと呼ぶことにします。）

MKII は，電源を入れたとき，あるいはリセットボタンを押したときは，0 番地から命令を実行し始めます。つまり，ROM に書き込まれている BASIC インタプリタが動き始めるわけです。　BASIC インタプリタは，まず，フロッピィディスクユニットが接続されているかどうかを調べ，接続されていなければ，第7章の ROM-BASIC モードを起動します。接続されていれば，フロッピィディスクユニットにセットされているシステムディスクを読みはじめます。

システムディスクに書き込まれている DISK-BASIC モードの拡張機能は RAM に読み込まれます。このようにして，DISK-BASIC モードが起動します。

8.2　N80 DISK-BASICとDISK-BASIC

MKII が，２つの DISK-BASIC モードを持っていることを，**第5章 ソフトウェア概説** で述べました。ひとつは，**N80DISK-BASIC,** もうひ

とつは PC-8001 のもつ **DISK-BASIC** です。5章で示した各 BASIC の機能包含図を，もう一度お目にかけます。

N$_{80}$-BASIC
DISK-BASIC
N-BASIC
C
A
B
D
N$_{80}$ DISK-BASIC

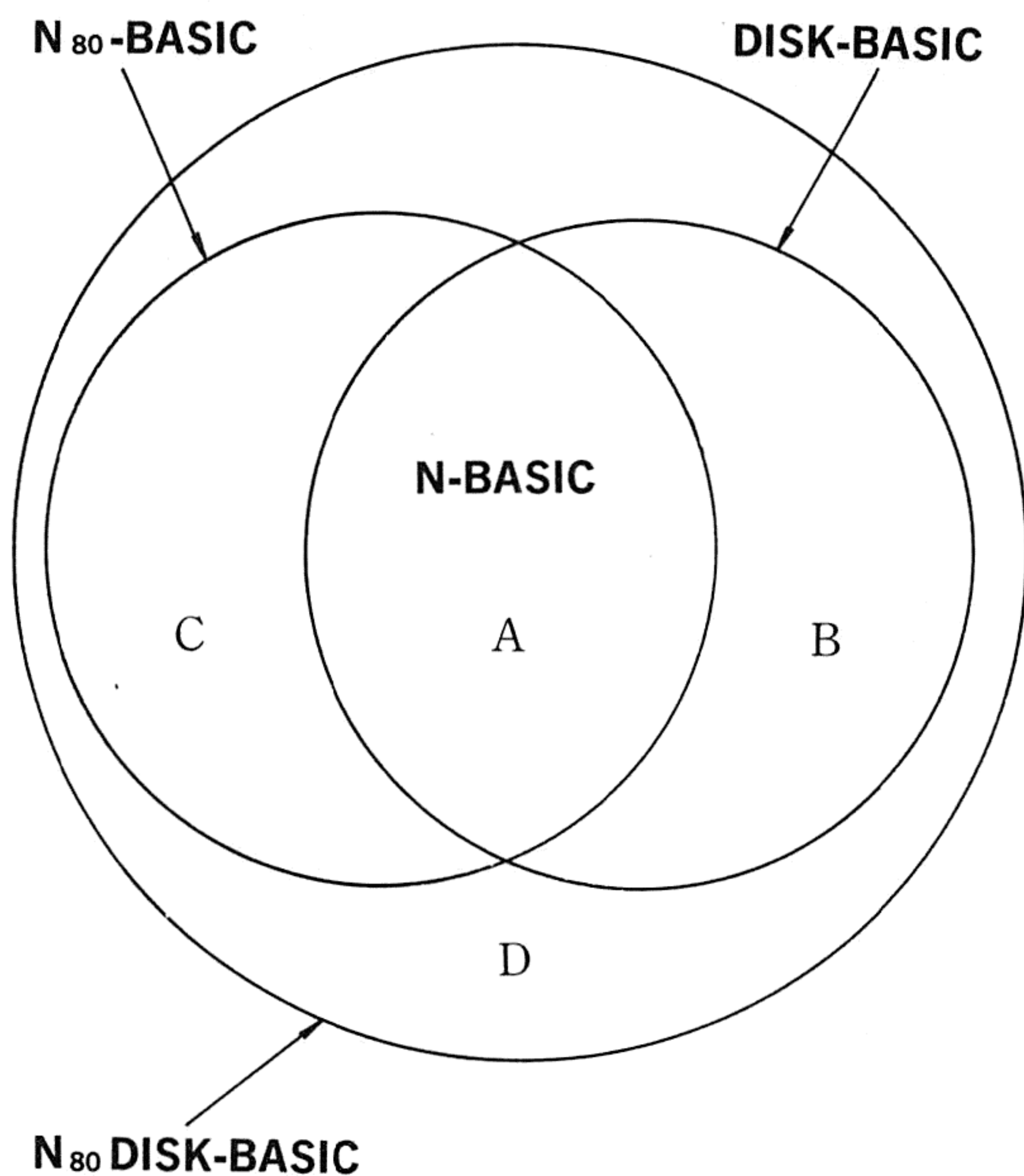



図８．１　４つのBASICモードの機能包含図

N$_{80}$DISK-BASIC と，**DISK-BASIC** の機能の包含関係は，図 8.1 のようになります。**N$_{80}$DISK-BASIC** は，**N$_{80}$-BASIC**（図 8.1 のAとCの部分）に，フロッピィディスクを扱うための命令や関数などの機能（図 8.1 のBとDの部分）を追加してできています。同じように **DISK-BASIC** は，**N-BASIC**（図 8.1 のAの部分）にフロッピィを扱うための命令や関数などの機能（図 8.1 のBの部分）を追加してできています。

この章では 8.1 から 8.4 までを除いて，すべて **N$_{80}$DISK-BASIC** についての説明をしていきます。**DISK-BASIC** の使い方については，ミニフロッピィディスクユニットなどに付いているユーザーズマニュアルなどをご覧ください。

N$_{80}$DISK-BASIC と，**DISK-BASIC** は，ほとんど同じですが，ちがう

ところを，次にあげます。

N80DISK-BASIC と DISK-BASIC のちがい

1．N80DISK-BASIC は，8インチと5インチの両方のフロッピィディスクを使うことができますが，DISK-BASIC は，5インチのフロッピィディスクしか使えません。
2．DISK-BASIC は，フロッピィディスクの使用開始宣言の MOUNT コマンドと，使用終了宣言の REMOVE コマンドが必要です。REMOVE コマンドを実行しないで，フロッピィディスクをとりだすと，ファイルが壊れてしまいます。

　N80DISK-BASIC は，MOUNT コマンド，REMOVE コマンドを必要としません。
3．N80DISK-BASIC では，機械語のプログラムを，CMD BLOAD, CMD BSAVEコマンドで読み書きすることができます。
4．当然ながら，DISK-BASIC には，N80-BASIC のグラフィックスなどの機能（図8.1のCの部分）がありませんが，N80 DISK-BASIC は N80-BASICの機能をすべて含んでいます。

8.3　DISK-BASICモードに必要なシステムディスク

DISK-BASIC モードの起動には，フロッピィディスクユニットとシステムディスクが必要です。

N80DISK-BASIC と DISK-BASIC の2つのモード，各種のフロッピィディスクユニットとそれらの組み合わせによって，システムディスクと，ドライブにつける番号がちがってきます。表 8.1，表 8.2 にフロッピィディスクユニットの組みあわせの種類と，それに必要なシステムディスク，ドライブ番号のつけ方をあげます。

表8.1　**N₈₀DISK-BASIC**のためのフロッピィディスクユニットの組み合わせ，ドライブ番号の割り当て，必要なシステムディスク

PC-8881		PC-8882		PC-80S31または PC-8031-2W		PC-80S32または PC-8032-2W		PC-8032-1W		PC-8031-1W		ディスクユニット ＼ 必要なシステムディスク（すべて別売り）
ドライブ 1	ドライブ 2	ドライブ 1	ドライブ 2	ドライブ 1	ドライブ 2	ドライブ 1	ドライブ 2	ドライブ 1	ドライブ 2	ドライブ 1	ドライブ 2	
1	2											**PC-8087**
1	2	3	4									
1	2			3	4							**PC-8087**または **PC-8037-2W**
1	2	3	4	5	6							
1	2	3	4	5	6	7	8					
1	2			3	4	5	6					
1	2							3	4			**PC-8087**または **PC-8037-1W**
1	2	3	4					5	6			
1	2	3	4					5	6	7	8	
1	2							3	4	5	6	
				1	2							**PC-8037-2W**
				1	2	3	4					
								1	2			**PC-8037-1W**
								1	2	3	4	

表中の1～8までの番号は，割り当てられるドライブ番号です。空白は，フロッピィディスクユニットを接続しないことをあらわします。

N₈₀DISK-BASIC では，8インチ，5インチ両方のフロッピィディスクユニットを接続することができます。

注　意　**PC-8881, PC-8031-2W, PC-8031-1W**には，**N₈₈ DISK-BASIC**または**DISK-BASIC**のシステムディスクがついていますが，**N₈₀ DISK-BASIC**には使えません。

PC-8031-2W, PC-8031-1W のユーザーズマニュアルには，**DISK-BASIC** について書かれていますが **N₈₀ DISK-BASIC** については書かれてありません。

表 8.1 に示したドライブ番号は，**N₈₀DISK-BASIC** で，フロッピィディスクに読み書きするときに使います。フロッピィディスクユニットを複数接続した場合は，8インチのフロッピィディスクに若い番号がわりあてられます。

表8．2　**DISK-BASIC** のための フロッピィディスクユニットの組み合わせ，ドライブ番号の割り当て，必要なシステムディスク

PC-80S31または PC-8031-2W		PC-80S32または PC-8032-2W		PC-8031-1W		PC-8032-1W		PC-8031-1V		PC-8031-1V +FD1		ディスクユニット ／ 必要なシステムディスク(PC-80S31/32以外は添付)
ドライブ 1	ドライブ 2	ドライブ 1	ドライブ 2	ドライブ 1	ドライブ 2	ドライブ 1	ドライブ 2	ドライブ 1		ドライブ 1	ドライブ 2	
1	2											**PC-8034-2W**
1	2	3	4									
				1	2							**PC-8034**
				1	2	3	4					
								1				**PC-8034-1V**
										1	2	

DISK-BASIC では，8インチのフロッピィディスクユニットを使うことはできません。接続されていても無視します。

表 8.2 にあげた **DISK-BASIC** のシステムディスクおよび，**DISK-BASIC** の使い方が書かれたマニュアルは，PC-80S31/32以外のフロッピィディスクユニットなどに添付されています。

8.4　DISK-BASICモードの起動方法

1．DISK-BASICモードの起動

DISK-BASIC モードの 起動は， 次の手順をふみます。

1．「3.2 周辺装置の接続」に従って，システムを接続します。

2．システムディスクにあわせて,ディップスイッチ8をセットします。

システムディスク	ディップスイッチ
PC-8087 **PC-8037-2W** **PC-8037-1W**	ON 1 2 3 4 5 6 7 8 (8: off)
PC-8034-2W **PC-8034** **PC-8034-1V**	ON 1 2 3 4 5 6 7 8 (8: on)

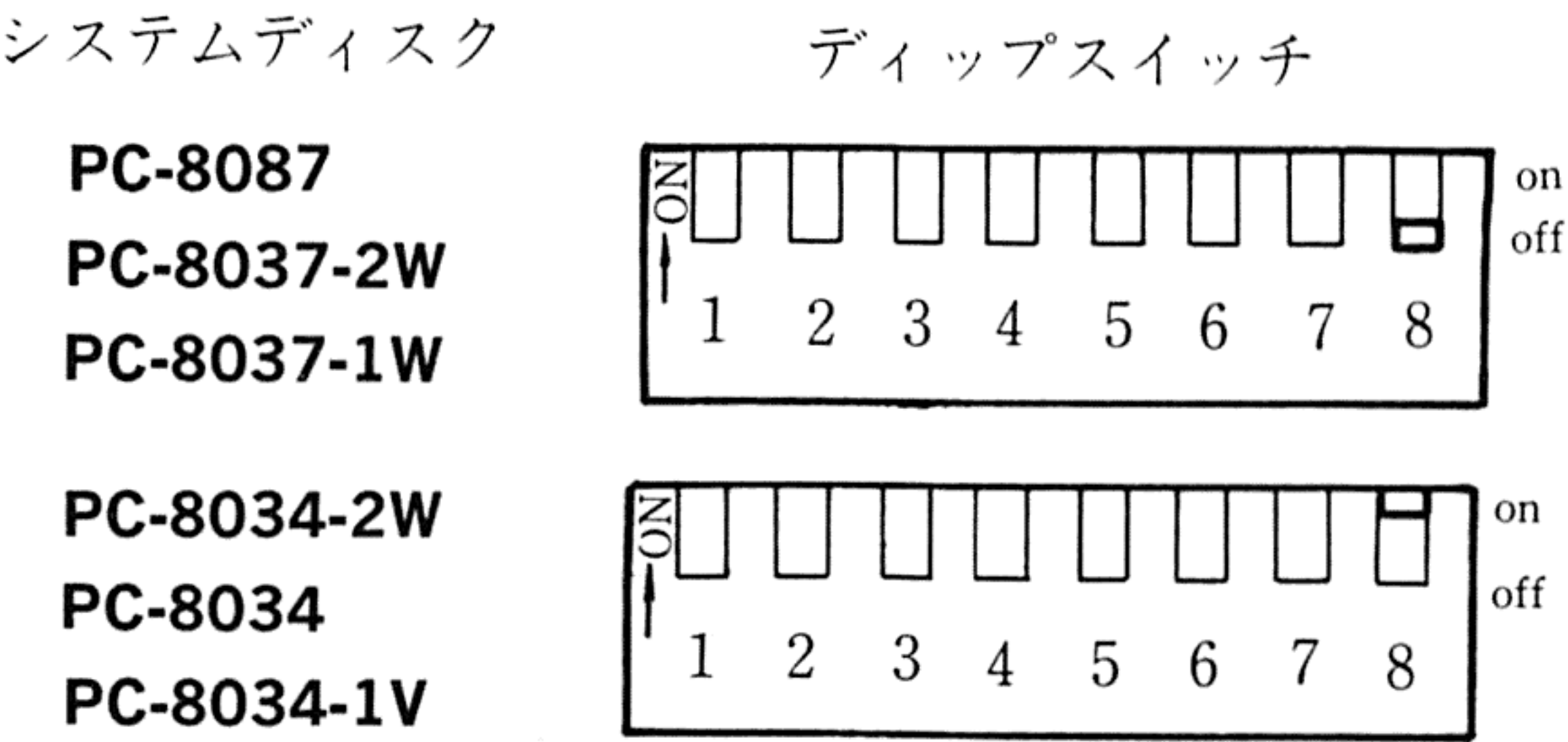



図8．2　システムディスクとディップスイッチ

N80DISK-BASIC のシステムディスクを使う場合は，ディップスイッチ８を off にします。 DISK-BASIC のシステムディスクを使う場合は，ディップスイッチ８を on にします。その他のディップスイッチは，on，off どちらでもかまいません。

（注　意）システムディスクとディップスイッチの組みあわせがまちがっていた場合，起動しますが，これは中途半端なモードですから使用しないでください。

３．フロッピィディスクユニットの電源を入れます。

４．システムディスクを，ドライブ１にセットして，ふたをしめます。８インチのフロッピィディスクユニットと５インチのフロッピィディスクユニットを両方ともに接続している場合は，８インチ，５インチどちらのドライブにセットしてもかまいません，（システムディスクはどちらか一方があればよいのです）。

５．MKIIの電源を入れて，リセットボタンを押します。するとフロッピィディスクユニットのモータが回転し始めます。

６．数秒後，ドライブ１のアクセス表示用 LED が点灯します。これはシステムプログラムの読み込みが始まったことを示しています。

（注　意）アクセス表示用 LED が点灯しているときは，フロッピィディスクに読み書きをしていることを示します。このときに，ドライブのふたをあけてフロッピィディスクをとりだしてはいけません。フロッピィディスクに傷をつけて，使えなくなってしまうからです。
ただし，拡張用ミニフロッピィディスクユニットのドライブ４は，他のドライブをアクセスしていないときは，常に LED が点灯しています。

７．さらに数秒たつと，LED が消え，画面に，次のメッセージが表示されます。

N80DISK-BASIC の場合

```
N-80 Disk version〔9−Feb−1983〕
How many files (0−15) ? □
```

DISK-BASIC の場合

Disk version 〔14−Nov−1980〕
How many files (0−15) ? □

または

Two surface disk version 〔20−Sep−1981〕
How many files (0−15) ? □

このメッセージの意味は 8.6 に述べます。ここでは [3] [RETURN] キーを入力します。続いてメッセージが現われます。

N₈₀DISK-BASIC の場合

```
N-80 Disk version 〔9−Feb−1983〕
How many files (0−15) ? 3
NEC N-80 BASIC ver 1.0
Copyright 1979(C) by Microsoft
Enhanced 1982 by NEC
Ok
□
```

DISK-BASIC の場合

How many files に続いて，次のように表示されます。

```
NEC PC-8001 BASIC Ver 1.3
Copyright 1979(C) by Microsoft

Ok
□
```

これで，**N₈₀DISK-BASIC** または **DISK-BASIC** が起動し，コマンド入力できる状態になりました。

(**注　意**) ここにあげたメッセージの中の，日付などの数字は，version によって異なることがあります。

N₈₀ DISK-BASIC または **DISK-BASIC**の起動がうまくいかない場合には，次のような原因が考えられます。

(1) **MKII** の電源を「on」にして，リセットボタンを押すと，すぐに ROM-BASIC モードのメッセージが画面に現われる場合。

MKII とフロッピィディスクユニットが接続されていない場合に，

このように動作することがあります。もう一度，接続をチェックしてください。

(2) **MKII**の電源を「on」にして，リセットボタンを押すと，フロッピィディスクユニットのアクセス表示用LEDが点滅し，次のようなメッセージが現われる場合。

＊＊＊ Two surface version ＊＊＊

＊＊＊ can't work on PC-8031-1W ＊＊＊

両面用のシステムディスク（**PC-8037-2W**)を，片面用のディスクユニット（**PC-8031-1W**)に入れて起動しようとすると，このようなメッセージが表示されて，起動は行われません。ディスクユニットと，システムディスクのタイプをあわせてご使用ください。

(3) **MKII**の電源を「on」にして，リセットボタンを押すと，数秒たってから，ROM-BASIC モードのメッセージが画面に現われる場合。

フロッピィディスクユニットの電源を調べてください。電源表示用LED が点灯していますか。電源表示用 LED が点灯しているにもかかわらず，このように動作する場合は，ミニフロッピィディスクユニットに取り付けた，接続ケーブルのコネクタが，逆に取り付けられている可能性があります。電源を「off」にして確かめてください。

(4) **MKII**の電源を「on」にして，リセットボタンを押すと，フロッピィディスクユニットのアクセス表示用 LED は点滅するが，しばらくすると，ROM-BASIC モードのメッセージが出力されるか，キーボードから何も入力できなくなる場合。

フロッピィディスクユニットにセットされているシステムディスクの選択がまちがっているか，あるいはその内容が壊れている場合にこのような動作をします。

2. システムディスクのバックアップ

N₈₀DISK-BASICまたは**DISK-BASIC**をはじめて起動したときは，必ずシ

ステムディスクのコピーをとる必要があります。システムディスクは一枚しかなく，誤った操作でシステムディスクをこわしてしまう可能性があるからです。フロッピィディスクのコピーを作ることをバックアップといいます。

バックアップはシステムディスクに入っているユーティリティプログラムを使って行います。**N₈₀DISK-BASIC**のバックアップユーティリティは“**backup N80**”です。 8.9.3節の手順に従ってバックアップしてください。 **DISK-BASIC**のバックアップユーティリティについては，ディスクユニットなどに添付してあるマニュアルをごらんください。

これからは，新しく作ったシステムディスクを使うことにして，もう一枚は大切にしまっておきましょう。

3．フロッピィディスクの初期化(フォーマッティング)

新しいフロッピィディスクを**N₈₀DISK-BASIC**または**DISK-BASIC**で使うためには，初期化(フォーマッティング)をする必要があります。前節でバックアップして作ったシステムディスクは，まずフォーマッティングし，その後データをコピーして作られたものです。このように，バックアップユーティリティでもフロッピィディスクのフォーマッティングをすることができますが，フォーマッティングだけを行うユーティリティも用意されています。**N₈₀DISK-BASIC**を使う場合は，8.9.1，8.9.2節を，**DISK-BASIC**を使う場合は，ディスクユニットなどに添付してあるマニュアルをごらんください。

> ここまでは，**N₈₀DISK-BASIC**と**DISK-BASIC**を同時に説明してきました。次節からは，**N₈₀DISK-BASIC**だけに絞って説明します。**DISK-BASIC**については，システムディスクや，フロッピィディスクユニットに添付されているユーザーズマニュアルをご覧ください。

8.5 プログラムのロード・セーブ

フロッピィディスクなしの **N80-BASIC** では，プログラムのロード，セーブはカセットテープを用いていましたが，**N80DISK-BASIC**では，これらの操作がフロッピィディスクを対象として行われます。このことは，ファイルの読み書きに要する時間を著しく短縮できるだけでなく，ファイルの信頼性がはるかに向上するため，安心して使用できます。

8.5.1 プログラムのセーブ

カセットテープにプログラムをセーブする場合，あなたは次のコマンドを使っていたはずです。

CSAVE 〈ファイル名〉

ところで **CSAVE** の "**C**"はカセットの**C**です。ここまで読んで，それならばフロッピィディスクへのセーブは **FSAVE** だろうと思い込んで試みた人は相当のあわて者です。しかし，われわれは，フロッピィディスクを使う前に少なくとも 8．1 から 8．5 のすべてを読んで，理解してくれることを願います。さもないと，あなたは大事なシステムディスクをこわしてしまうかもしれません。

フロッピィディスクに対するプログラムのセーブは次のように行います。

(書 式) **SAVE 〈ファイル名〉**

(Sample)
```
save "prog1"
save "prog1.bas"
save "2 : prog1"
```

この例の意味は明らかでしょう。プログラムに，**prog1** というファイル名をつけてフロッピィディスクにセーブします。

2番目の例の，.bas は拡張子です。

3番目の例の，2: はドライブ番号です。

ここでファイル名の一般形を述べておきましょう。

"ドライブ番号 ：ファイル名．拡張子"

ドライブ番号は1から最大8までです。

正しくないドライブ番号を指定すると

Bad drive number

のエラーが発生します。

ドライブ1のときドライブ番号は省略できます。ドライブ番号を省略する場合には：も省略します。

拡張子はファイルの型を指定します。これは単にファイル名の一部分と考えておけばよいでしょう。拡張子は書かなくてもかまいません。拡張子は3文字以下からなります。

ファイル名は**6文字以下**（ドライブ番号と拡張子を含まない）です。

この他に，**N80DISK-BASIC** では，機械語のプログラムやデータをフロッピィディスクにセーブすることもできます。機械語のプログラムのセーブには**CMD BSAVE**を使います。

(書　式) **CMD BSAVE <ファイル名>,<開始番地>,<プログラムのバイト数>**

(Sample) **cmd bsave "mprog", &ha000, &h20**

これは，A000H 番地から A019H 番地の機械語（長さ 20H）を **"mprog"** というファイル名でフロッピィディスクにセーブする例です。

(注　意) **N80DISK-BASIC** では，ドライブ番号がファイル名に含まれます。このマニュアルで，単に〈ファイル名〉と記述されている部分は，もしそのファイルがドライブ2を対象とするならば，すべてドライブ指定(2:) を含まなければなりません。

このことは特に後述する **KILLコマンド**において重要です。ドライブ1と2に同一のファイル名を持つファイルがある場合でドライブ2にあるファイルを削除したい場合，ドライブ指定を忘れて，ドライブ1にあるファイルを削除してしまうことがあり得ます。

実際,このドライブ指定は非常に忘れやすいので，十分に注意してください。

8.5.2 ファイル名のリストをとる

フロッピィディスクに格納されているファイル名のリストをとるときには次のコマンドを使います。

(書　式) **FILES<ドライブ番号>**

〈ドライブ番号〉を省略するとドライブ1が指定されます。

フォーマッティング直後の，何も記録されていないフロッピィディスクに対し **FILES** を実行したときには1行の空白のみが表示されます。

次の例は **SAVE, CMD BSAVE**および**FILES**を行っています。

(Sample)
```
save "prog1"
Ok
cmd bsave "mprog1", &ha000, &h20
Ok
files
prog1  .  1      mprog  *  1
Ok
```

ファイル名のリストをプリンタに出力することもでき，これには次のコマンドを使います。

(書　式) **LFILES 〈ドライブ番号〉**

PRINT と **LPRINT** の関係と同じです。

なお，前例で **prog1** のあとに小数点が表示されています。この小数点は，そのファイルがバイナリ形式で作られたことを示します。 8.5.4 で詳しく述べるつもりですが，プログラムをセーブするときには，圧縮した形でのバイナリ形式のファイルと，アスキー形式のファイルの2つの形式を選ぶことができ，アスキー形式のファイル（データファイルはすべてアスキー形式です）では，ファイル名の6文字目と7文字目の間に小数点ではなく空白が表示されます。また機械語のファイル(**CMD BSAVE**でセーブされたファイル)は，6文字目と7文字目の間に“＊”が表示されます。これは単に **FILES** で表示させたときだけの話で，ファイル名としては，そのファイルを作ったときにつけた名前を用いることは言うまでもありません。

たとえば，“**abcdefg**”という名前のファイルを作ると，**FILES** で，

abcdef. g （バイナリセーブのとき）
abcdef　g （アスキーセーブのとき）
abcdef＊g （機械語のとき）

と表示されますがこれをロードするときに，小数点は入れても入れなくてもかまいませんが，空白や＊は入れてはいけません。ふつうは，そのファイルを作ったときと同じように，“abcdefg”というファイル名で呼びます。

8.5.3 プログラムのロード

SAVE の説明を読んだ人はおそらくプログラムのロードに使う命令が何かわかったはずです。

(書　式) **LOAD 〈ファイル名〉**

(Sample)
```
load "prog1"
load "prog1.bas"
load "2 : prog1"
```

最後の例に示すとおり，ドライブ１以外のドライブからロードする場合，ドライブ番号をつける必要があります。これは忘れやすいから注意してください。

CMD BSAVEでセーブした機械語のロードには，**CMD BLOAD**を使います。

(書　式) **CMD BLOAD 〈ファイル名〉〔,〔ロードアドレス〕〔,R〕〕**

(Sample)
```
cmd bload "mprog"
cmd bload "mprog", &ha000
cmd bload "mprog", , r
```

１番目の例は，“**mprog**”というファイル名の機械語を **CMD BSAVE**でフロッピィディスクにセーブする際に指定した開始番地からロードします。
２番目の例は，ロードする際，**A000H**番地から機械語をロードします。
最後の例は，１番目の例と同様にプログラムをロードしたあとに，**CMD BSAVE**で指定した開始番地またはロードアドレスからプログラムをスタートさせます。

データファイルをロードしたら？

このことは規定されていません。最悪の場合は，リセットをかけてもう

一度始めから，N_{80}DISK-BASIC をスタートさせることになります。

この種のトラブルを避けるには，ファイル名でデータかプログラムを区別させるしかありません。その場合はファイルの拡張子をうまく利用して下さい。

(Sample) プログラムのファイルには必ず（".N80" をつける。
省略する。

データファイルには ".dat" をつける。

機械語のファイルには ".bin" をつける。

8.5.4 アスキーセーブとMERGE

SAVE でプログラムをセーブしたとき，フロッピィディスクには，プログラムが圧縮された型で書き込まれます。

普通プログラムをセーブするには，この **SAVE** で行うのが，フロッピィディスクのスペースの有効利用，およびロード，セーブに要する時間の点で最も有利です。ところが，ある特殊な目的で，この圧縮した型での書き込みを行わず，プログラムのリストをとったときと同一の型式でフロッピィディスクにセーブすることがあります。この方法が**アスキーセーブ**であり，主な目的がその次に述べる**マージ**です。

アスキーセーブ

アスキーセーブの方法は非常に簡単です。

(書　式) **SAVE〈ファイル名〉, A**

最後の，", A"がアスキーセーブを指示します。

マージ

メモリ内のプログラムと，ディスク内のプログラムを行単位で合成して，1つのプログラムにすることをマージすると言います。これを行うコマンドが **MERGE** なのです。

(書　式) **MERGE〈ファイル名〉**

MERGE は，メモリ内にあるプログラムと，指定したディスク内のプ

ログラムファイルを結合する命令です。合成されたプログラムはメモリ内にはいります。

MERGE を行うファイルはアスキー型式のファイルであることが要求されます。すなわち，",A"（**A**オプション）付きでセーブされたファイルでなければなりません。

8.5.5 ファイルの削除

ファイルの削除は **KILL** で行います。

（書　式） **KILL <ファイル名>**

（Sample） **kill "Prog1"**
kill "2 : fileA"

ところで，**N_{80}DISK-BASIC** においては，あるファイルを作成するとき，すでに同一名のファイルが存在している場合には，そのファイルを自動的に削除し，同じ名前の新しいファイルを作ります。したがって，同一ファイル名のファイルを書き込む場合には，旧ファイルを **KILL** する必要はありません。

もっとも，この方式も良し悪しで，うっかりすると，ある重要なファイルと同一名で別のファイルを **SAVE** する危険があるからで，これを防ぐには，重要なファイルには「**書き込み禁止属性**」をつけておくのが安全です。これには，次項で述べる **SET** を用います。

8.5.6 ファイルの属性

ファイルの属性には，次の2種が指定できます。

(1) **書き込み禁止**

(2) **read after write**

通常，何も指定しなければ，どちらの属性も持っていません。特に重要なファイルで，以後書き込みも変更しない予定のファイルには，(1)の書き込み禁止属性を指定しておくと良いでしょう。指定したあとで，気が変わ

ってそのファイルの内容を変更したくなったら，その属性を解除して，しかるべきファイルの内容を変更して，その後もう一度，書き込み禁止属性を指定すればよいのです。

さて，これらの属性の指定，解除には **SET** を用います。

SET の使い方にはさまざまのバリエーションがありますが，最も良く使うであろうそれは，次の形式です。

書式 **SET 〈ファイル名〉,〈属性文字〉**

属性文字は**R**，**P**，**それ以外**であり，それぞれ

R…………read after write
P…………書き込み禁止
それ以外…属性の解除

Sample **set "prog1", "P"**

この例は，prog1 というファイル（すでにフロッピィディスク上に存在していることが要求されます）に，書き込み禁止属性をつけます。

Sample **SET "prog1", ""**

この例は，そのファイルのすべての属性を解除します。

注意 属性文字**R**，**P**は必ず**大文字**をわなければなりません。小文字を使うと，「属性の解除」とみなされます。

次に，他の形式の **SET** の使い方を紹介しましょう。これは読みとばしてもかまいません。

書式 **SET 〈ドライブ番号〉,〈属性文字〉**

この形式はそのドライブに入っているフロッピィディスクの属性を指定します。以後作られるファイルはすべて，その属性を持ちます。

1枚のフロッピィディスク全体の属性は8.7.6 で述べるIDセクタの最初の1バイトの値で設定されます。

書式 **SET ＃ 〈ファイル番号〉,〈属性文字〉**

この形式は指定したファイル番号で読み書きをするデータファイルの属性を指定します。ただし，これはそのファイルがオープンされている間だけ有効であり，ディレクトリには記録されません。

参照 ファイル番号，オープンなどの用語は8.6節で説明します。

8.5.7 ファイル名の付け換え

すでに存在しているファイルのファイル名は，**NAME** により別の名前に変更することができます。

(書　式) **NAME 〈旧ファイル名〉 as 〈新ファイル名〉**

(Sample) **name "oldfil" as "newfil"**

(注　意) 旧ファイル名と新ファイル名のドライブ番号は同じでなければなりません。

(Sample) 正しい　**NAME "DEMO" AS "SOFT"**

正しい　**NAME "2:TEST" AS "2:WORK"**

誤り　**NAME "2:WORK" AS "NEC"**

8.5.8 Rオプションと RUNコマンド

まず，**RUN コマンド**から話しましょう。**N$_{80}$DISK-BASIC**では，フロッピィディスク用に**RUN**コマンドの機能が強化されており，次の形式が使えるようになりました。

(書　式) **RUN 〈ファイル名〉**

この形式の **RUN** コマンドは指定したファイルをロードして，それを実行します。たとえば，

RUN "prog1"

はドライブ1にある **prog1** という名前のプログラムファイルをロードし，ロード終了後そのプログラムを実行します。すなわち，これは次の2つのコマンドを続けて実行したのと同じ意味です。

LOAD "prog1"

RUN

次に**Rオプション**について述べてみます。Rオプションは**LOAD, CMD BLOAD** および **RUN コマンド**に付加でき，次の形式を持ちます。

(書　式) **LOAD 〈ファイル名〉, R**

CMD BLOAD 〈ファイル名〉, R

RUN 〈ファイル名〉, R

LOAD の **R** オプションと **RUN** の **R** オプションの機能は全く同一です。すなわち，指定したファイルをロードし，それを実行します。この時，それ以前にオープンされているファイルはクローズされません。**R** オプションなしの **LOAD, RUN** コマンドは，すべてのオープンされているファイルをクローズします。これらの **R** オプション付きコマンドは，プログラムのチェインを行うときに便利です。

なおプログラムのチェインについては **8.5.9**に，オープン，クローズについては **8.6**に述べられています。

8.5.9 プログラムのチェイン

プログラムが大きくて，全部がメモリにはいりきらない場合，プログラムをいくつかのモジュールに分けておいて，それぞれ別のファイル名を付けてフロッピィディスクにセーブしておき，必要なモジュールのみをロードして実行し，その後，自動的に次のモジュールをロードして実行を続けるという方法が使われます。このような方法のことを，**プログラムをチェインにする**，と呼びます。このときに，前項に述べた **R** オプション付きのコマンドが便利に使えます。 **N80-BASIC , N80 DISK-BASIC**では，すべてのコマンドがプログラム内に書けるのです。

図 8.3 はチェインの方法を簡単に説明しています。

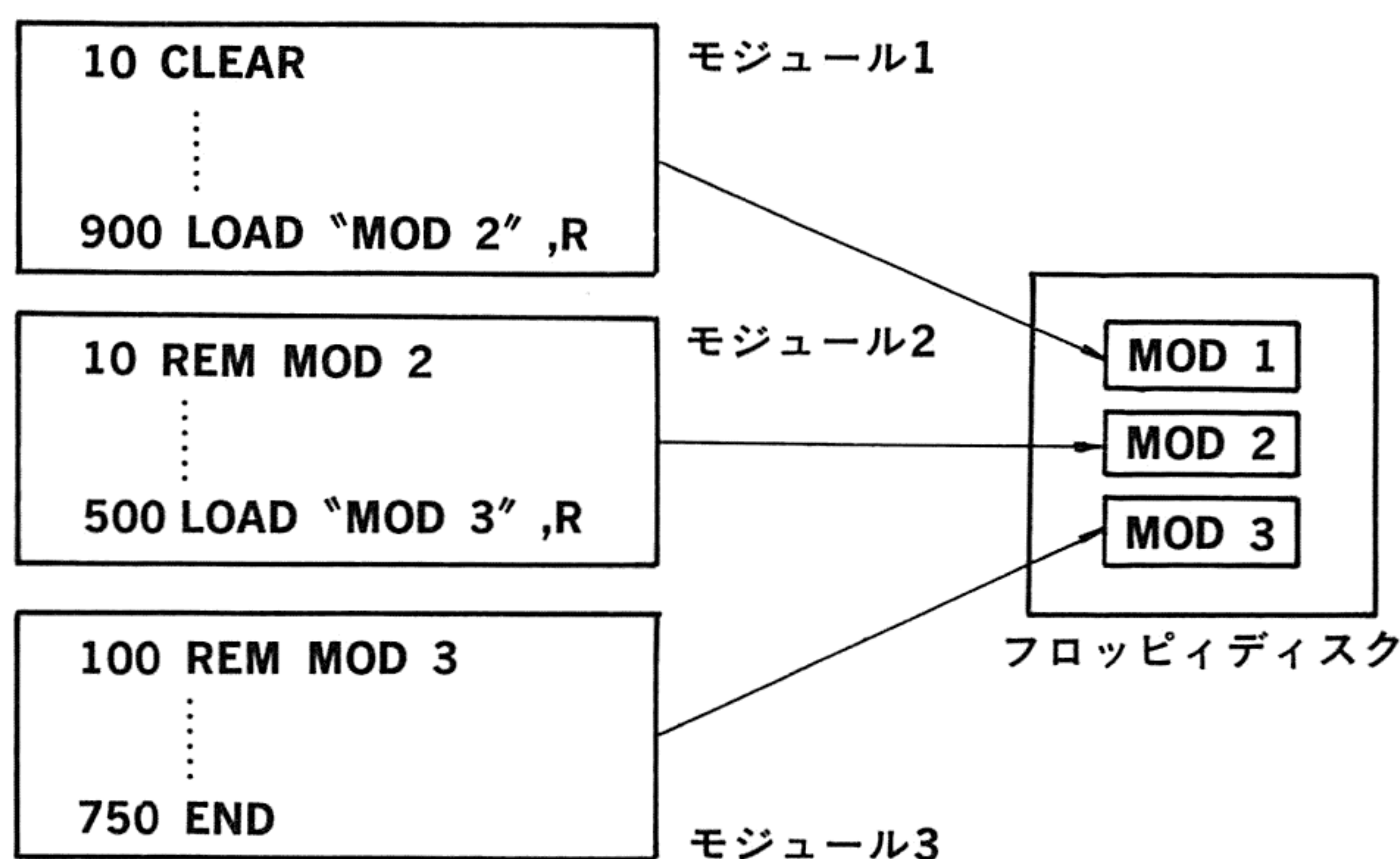



図8.3　プログラムのチェイン

このように各モジュールの最後のステートメントが，次のモジュールをロードし，それを実行させます。モジュール3は最後に実行されるので，もはや次のモジュールを起動することはありません。

このプログラムを実行させるときにはモジュール1のみをロードし，実行させます。モジュールは，その実行を終えると，自動的に自分自身をクリア（**NEW**）し，モジュール2をロードして，それを実行します。すなわち，プログラム実行中は，常にたった1つのモジュールだけがメモリ上にあります。非常に大きなプログラムでもこのように小さなモジュール群にわけることで，実行させることができるのです。

プログラムをチェインするときには次の点を理解しておいて下さい。

(1) 2つのモジュールに同一の行番号があっても，それらは全く別個のものです。このことは，あるモジュールをロードするときには，それ以前にメモリ上にあったモジュールは **NEW** されることから，容易に理解されるでしょう。

(2) あるモジュールから他のモジュールの行を参照することはできません。すなわち，あるモジュールの内から，他のモジュールの行へ **GOTO** することや **GOSUB** することはできません。逆に，このようなことのないように各モジュールを分離すべきです。

(3) 次の点は重要です。あるモジュールから他のモジュールに変数の値をわたすことはできません。あるモジュールをロードしたとき，その時の変数の値はすべてクリアされます。したがって，モジュール間で変数データのうけわたしを行うときには，次の例のように，ディスクファイルを介して行います。

Sample モジュール **MOD1** から **MOD2** に，変数**A**，**B**，**C$**を渡すことを考えてみましょう。

```
10 REM  モジュール MOD 1
         :
100 OPEN "VAR" FOR OUTPUT AS #1
110 PRINT #1, A, B
120 PRINT #1, C$
130 CLOSE #1
140 LOAD "MOD 2" ,R
```

MOD 1

```
10 REM モジュール MOD 2
20 OPEN "VAR" FOR INPUT AS #1
30 INPUT #1, A, B
40 LINE INPUT #1, C$
50 CLOSE #1
         :
```

MOD 2

図８．４　プログラムの受け渡し

以上の点から想像できるように，プログラムをチェイン構造にすることは，ある程度の実行効率の低下をひきおこします。特に，モジュール間でひきわたすべき変数の数が多いときにはこれが顕著にあらわれるでしょう。したがって，プログラムのチェイン化はむやみやたらと行うべきではありません。

なお，(3)で述べた変数のうけわたしについては，データファイルを用いています。データファイルの詳細については次節 8.6 に記述されています。

8.6　データのロード・セーブ

8.6.1　シーケンシャルファイルとランダムアクセスファイル

N_{80}DISK-BASIC のデータファイルには，**シーケンシャルファイル**と**ラ**

ンダムアクセスファイルがあります。

シーケンシャルファイルとは，連続的に書かれたデータファイルのことで，書いた順序でしか読めないファイルです。カセットテープに作られるファイルは，このシーケンシャルファイルです。

シーケンシャルファイルの特徴として，次の事があげられます。

(1) データのロード，セーブを行う場合は必ずファイルの先頭から順次行います。
(2) ファイル内の一部のデータだけを書き換えることはできません。
(3) ファイルの変更は，ファイルの最後に新たなデータを追加することだけです。
(4) 比較的簡単に使えます。
(5) １つ１つのレコード（データ処理の単位で，レコードの集りをファイルと呼びます）の大きさは，255 文字以内で自由に変えられます。

これに対して，ランダムアクセスファイルはシーケンシャルファイルの不得意とするファイル内の任意のデータ（レコード）を直接アクセスすることを得意とするファイルです。

ランダムアクセスファイルの特徴は，

(1) ファイル内の各レコードにランダムにアクセスできるので，データ（レコード）のロード，セーブを行なう場合はファイルの先頭から，順次行なわれなくてもよいのです。
(2) ファイル内の一部のデータだけを書き変えることができます。
(3) ファイルの最後に，新たなデータを追加することもできます。
(4) シーケンシャルファイルに比べて，プログラムが複雑になります。
(5) １つ１つのレコードの大きさは，256 バイトに固定されています。

シーケンシャルファイルを使いこなせば，ほとんどのデータ処理ができますが，ランダムアクセスファイルを使いこなしてこそ，コンピュータのデータ処理機能をフルに活用したと言えるでしょう。

8.6.2 ファイルのOPENとCLOSE

カセットデータファイルとの入出力では **INPUT#-1** と **PRINT#-1** を用いました。フロッピィディスクに対するデータの入出力も，シーケンシャルファイルの場合は **PRINT#**およびその親類の **LINE INPUT#，PRINT# USING**を使います。

(備　考) ランダムアクセスファイルでは**GET#**と**PUT#**を使います。

ところでカセットデータファイルを作るときには，ファイル名を使いませんでした。一方，ディスクファイルはすべてファイル名により識別されます。したがってディスクに対するデータの入出力を行うには，その前にシステムに対して，入出力の対象となるファイルの名前を知らせておく必要があります。これが **OPEN** です。またそのファイルに対する読み書きが終了したら，システムに対して，その旨を連絡しておくことが要求されます。これが **CLOSE** です。

8.6.2.1 OPEN

OPENにより，ユーザはシステムに対して次の3つのことを知らせます。

(1) これから入出力を行うファイルの名前

(2) そのファイルに対して何を行うのか（入力か，出力かなど）

(3) 今後，そのファイルを何と呼ぶか

(3)は少しわかりにくいかもしれません。次の例にもとづき説明しましょう。

(Sample) **OPEN "data1" FOR INPUT AS #1**

```
OPEN "data1" FOR INPUT AS #1
      ↑          ↑         ↑
     (1)        (2)       (3)
```

data1 がファイル名であり，このファイルに対して入力を行う（**FOR INPUT**）。

AS#1 が(3)の部分です。実際に **INPUT#** や **PRINT#** で入出力を行うとき，何らかの形でこれらの文の対象となるファイルを指定しなければなりません。１つのプログラムの中でたった１つのデータファイルだけを相手に入出力を行うとはかぎらないからです。通常，複数のファイルが入出力

の対象となります。**INPUT#**や，**PRINT#**でファイル名を書くことによって対象を指定することも考えられます。ところがこれはかなり煩雑です。したがってファイル名に適当な番号をつけて，以後 **PRINT#**，**INPUT#** ではファイル名のかわりにこの番号を使うことにします。(3)で書いた番号がこの番号です。

まとめてこの例を日本語になおしますと

「**data1** という名前のファイルを入力用に使います。今後，（このプログラム内では）このファイルを単に1番のファイルと呼ぶことにします。」

この(3)の番号のことを**ファイル番号**と呼びます。また，**OPEN** によるファイルの使用宣言を行うことを**ファイルを開く，オープンする**と呼ぶことにしましょう。

ファイル番号は，そのプログラム内でのみ通用する仮のファイル名であり，異なるプログラムで，そのファイルを別のファイル番号でオープンしてもかまいません。

データファイルの入出力を行うには，必ずそのファイルを **OPEN** しておかなければなりません。これを忘れると **File not OPEN** のエラーが発生します。

ところでファイル番号としてどのような数でも許されるのでしょうか？

答えは NO です。まず，ファイル番号は1以上の整数でなければなりません。また，**N_{80}DISK-BASIC** の起動時の，**How many files?** で入力した数以下でなければなりません。

今や，**How many files?** という質問の意味が明らかにされます。このメッセージは「同時にオープンされるファイルの数」を問いあわせているのです。

8.4 では，深く悩まず**3**を入力するように指示しました。あなたがよほどのへそまがりでないかぎり（プログラマのほとんどはへそまがりなのですが），**3**を入力してくれたと思います。その場合，あなたはシステムに対して，同時にオープンするファイルの数は3本であり，したがってファ

イル番号としては，1，2，3を使うことを宣言したことになります。

ファイルは同時に 15 本オープンすることができます。それならば，何も「3本だ」などとせこいことを言わずに景気よく 15 を指定すればよいと思う人もいるでしょう。やってみるとわかりますが，同時にオープンしうるファイルの数を増やせば増やすほど，プログラムに使える RAM の容量は減るのです。（**? FRE（0）** でわかる）これは，ファイルを1本オープンするたびにそのファイル用の入出力バッファ（ディスクとの読み書きをするデータを一時的に蓄わえておく場所）を必要とするからです。

ほとんどの応用では同時にオープンするファイルの数は3本まででしょうから，例の質問に対しては3と答えておけばよいのです。

ところで，「同時にオープンするファイル」という言葉は明確には説明されておりません。これについては **CLOSE** の説明の時に明らかにしましょう。

一方，**OPEN** の例の(2)の部分，すなわちそのファイルに対して何を行うのかを指定する部分ですが，ここには次のうちのどれかを書きます。

(1) INPUT
(2) OUTPUT
(3) APPEND
(4) 省略

(1)と**(2)**は明らかです。**(3)**は追加です。この節の最初に，シーケンシャルファイルに対する変更は追加のみが可能であると述べました。**(3)**はこの追加処理を指定します。**(4)**は主としてランダムアクセスファイルを扱うときに用いる指定ですが，シーケンシャルファイルにも使うことはできます。

8.6.2.2 CLOSE

ファイルに対して読み書きをするにはまず **OPEN** でそのファイルに対して使用宣言をしました。そのファイルに対する読み書きが終了したら，システムに対して，その旨を連絡しておくことが要求されます。これが **CLOSE** です。すなわち，**CLOSE** はファイルの使用終了を宣言します。

あるファイルに対して **OPEN** を実行してから，それを **CLOSE** する

まで，そのファイルはオープンされています。

プログラムのある時点で **OPEN** されているファイルの本数が，前節で述べた，「同時にオープンするファイル」の数のことです。

1つのプログラムで，多数のファイルをアクセスする場合でも同時にオープンされているファイルの数はそう多くないことが多いものです。

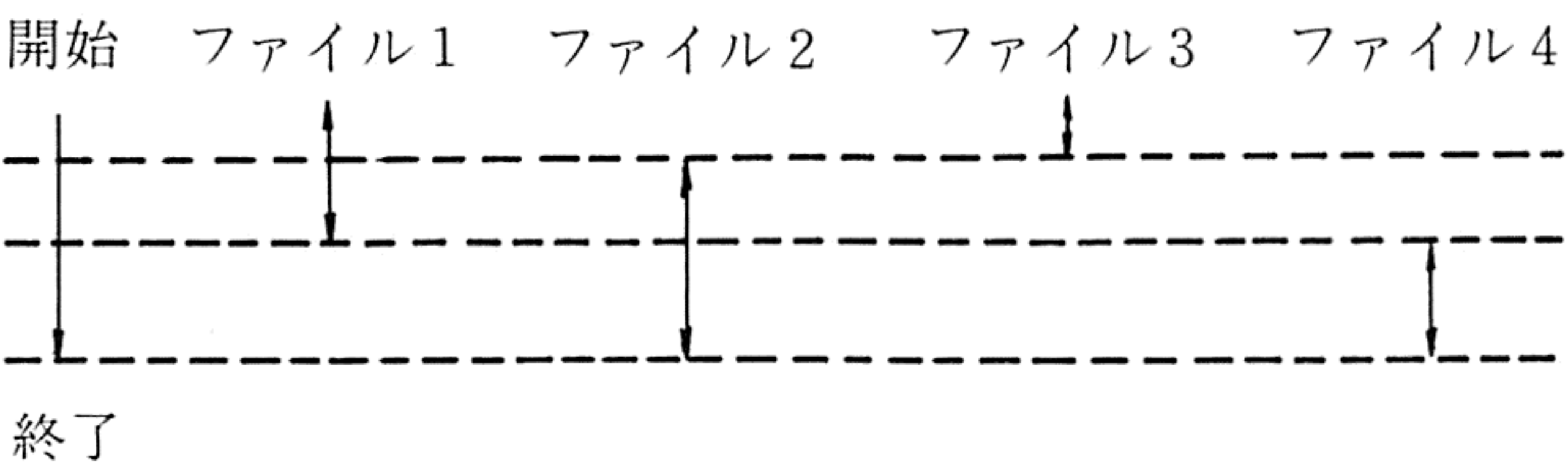



図8.5　プログラムの流れと各ファイルの使用期間(その1)

この例では，プログラム全体で4本のファイルをアクセスしますが，同時にオープンされるファイルの最大数は2本なのです。したがってこの場合でも，　**How many files?** には**3**と答えておけばおつりがくるというものです。

ところが，この場合，使用できるファイル番号は**1**から**3**まであり，また同時にオープンされるファイル番号はすべて異なっていなければなりません。

このようなとき，**CLOSE** で，すでにすべての予定されたアクセスの完了したファイルの使用終了宣言をだしておけば，そのファイル番号を別のファイルに使えます。

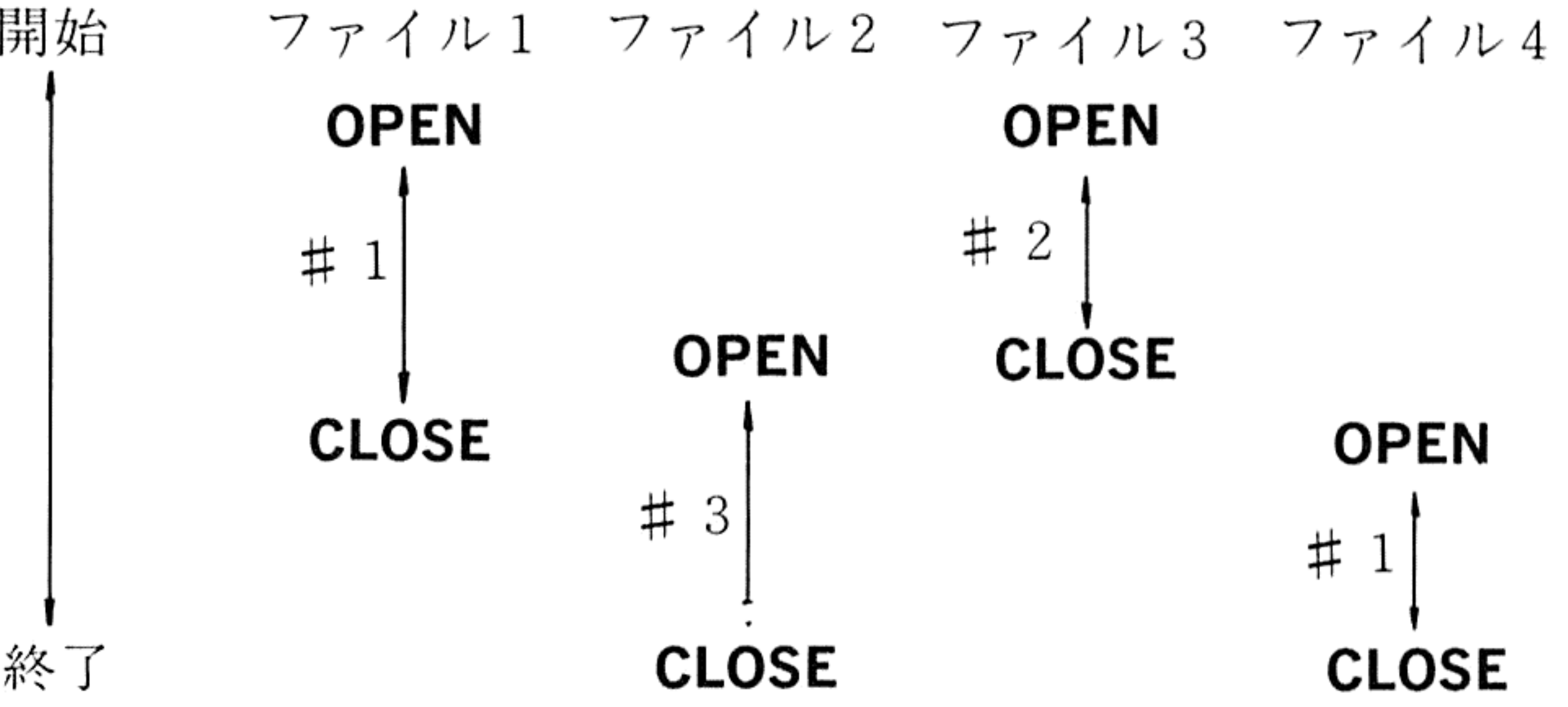



図8.6　プログラムの流れと各ファイルの使用期間(その2)

この例では，ファイル1とファイル4は同じファイル番号#1を使っています。しかし，これら2つのファイルは全く別のものとして扱われるのです。ファイル番号とファイルとの結びつきは，そのファイルが **OPEN** されている間だけ存在します。そのファイルが **CLOSE** された時点で結びつきがなくなるのです。

CLOSE の目的と意味はこれで説明されました。次にその構文を説明しましょう。

(書　式) **CLOSE**〔〈**ファイル番号**〉〕〔，〈**ファイル番号**〉〕

(Sample) **CLOSE**

CLOSE1

CLOSE#3

CLOSE1，2，3

最初の例はすべてのファイルをクローズします。

2番目と3番目の例はそれぞれファイル番号**1**，**3**のファイルのみをクローズします。このように，ファイル番号は#をつけなくてもかまいません。

最後の例はファイル番号**1**，**2**，**3**の3つのファイルをクローズします。

CLOSE がない場合でも，自動的にファイルが **CLOSE** される場合があります。

(1) **END** を実行したとき

(2) プログラムを変更したとき

(3) **RUN，LOAD** を実行したとき

(注　意) **OPEN** したファイルは，必ず **CLOSE** しなければなりません。もし **CLOSE** をしないうちにフロッピィディスクを取り出すと，ファイルがこわれてしまいます。

ここでシーケンシャルファイルを作る，簡単なプログラム例を示しましょう。

これはキーボードから入力された文字例を **end** が入力されるまで，

そのままの形でディスクに書きます。次にそのファイルを読み，数字をすべて＊に変換し，それを別のファイルに書きます。最後に，変換ずみのファイルを読み込み，それを表示します。

特に何の役にたつ，というものではありませんけど，フロッピィディスクファイルの使い方の参考にはなるでしょう。

```
list
10 '
20 ' data トウロク
30 '
40 OPEN "data" FOR OUTPUT AS #1
50 LINEINPUT "?"; A$
60 IF A$="end" THEN 90
70 PRINT #1, A$
80 GOTO 50
90 CLOSE #1
100 '
110 ' ヘンカン
120 '
130 OPEN "data" FOR INPUT AS #1
140 OPEN "data1" FOR OUTPUT AS #2
150 IF EOF(1) THEN 230
160 LINEINPUT #1, A$
170 IF LEN(A$)=0 THEN 210
180 FOR I=1 TO LEN(A$)
190 IF MID$(A$, I, 1)>="0" AND MID$(A$,I,1)<="9" THEN
MID$(A$,I, 1) ="*"
200 NEXT I
210 PRINT #2, A$
220 GOTO 150
230 CLOSE 1, 2
240 '
250 ' ヒョウジ
260 '
270 OPEN "data1" FOR INPUT AS #1
280 IF EOF(1) THEN 320
290 LINEINPUT #1, A$
```

```
300 PRINT A$
310 GOTO 280
320 CLOSE #1
330 END
Ok

run
?abc123defgh789ijklmn
?o0pq678rstu338. . .
?12345678
?zxcv b44 4ff
?end
abc***defgh***ijklmn
o*pq***rstu***. . .
********
zxcv b** *ff
Ok
```

8.6.2.3 OPEN, CLOSEの別のみかた

この節はすでに説明した **OPEN, CLOSE** の，別のみかた，すなわちユーザ側から見たみかたではなく，むしろ インタプリタ 側からのそれを説明しましょう。興味がなければ，この項は無視して，次項に進んでもかまいません。

ファイルポインタ

インタプリタはファイルの どこをアクセスしているかを示すポインタを **OPEN** されているファイルごとに1つずつ持っています。

OPEN はこのファイルポインタの初期設定を行うものと考えることができるわけです。

OPEN for input

ファイルポインタをそのファイルの先頭に設定します。指定したファイルはすでに存在していなければなりません。

OPEN for output

ファイルがあればそのファイルの先頭にファイルポインタを設定しま

す。ファイルがなければ新たにそのファイルを作って（名前を登録して）ファイルポインタをそのファイルの先頭に設定します（同じ名前のファイルはこの時点でこわされます）。

OPEN for append

ファイルポインタをそのファイルの最後のレコードの直後に設定します。

OPEN（モード省略）

ファイルがあればそのファイルの先頭にファイルポインタを設定し，なければ新たに作ってその先頭にファイルポインタを設定します。

ディスクバッファ

フロッピィディスクは1バイト単位での読み書きはできません。最小の入出力単位は1セクタです。 したがって， 1セクタ分の バッファを用意しておき，そのバッファに対してバイト単位の読み書きを行います。

OPEN は，そのファイルが使うディスクバッファを用意します。

ディスクバッファは **OPEN** されているファイル毎に1本ずつあります。一度あるファイルに与えられたファイルバッファはそのファイルが **CLOSE** されるまで他に転用できません。 **CLOSE** はそのファイルが専用していたバッファをとりもどし，他のファイルが使用できるように確保しておきます。

システムが用意するディスクバッファの総数は， **N80DISK-BASIC** の起動時に **How many files?** にユーザーが応答した数です。

8.6.3 シーケンシャルファイルにおける データのロード・セーブ

シーケンシャルファイルにおける データの ロード・セーブは **INPUT＃**, **PRINT＃**およびそれらのバリエーションである **LINE INPUT＃**, **PRINT＃ USING**で行ないます。

データのロード命令は， **INPUT＃**, **LINE INPUT＃**で, セーブ命令は **PRINT＃**, **PRINT＃ USING**です。

シーケンシャルファイルは 255 文字までの自由な長さのレコードをCR/LF で区切って，つないでできます。

ここでは，シーケンシャルファイルにおけるデータのロード・セーブとデータの追加について，プログラム例をあげながら説明していきましょう。

次のプログラムをキーボードから入力してください。

プログラム	説明
100 OPEN 〝DATA〟 FOR INPUT AS ＃1 RETURN	ファイルを開きます。
200 INPUT 〝エイタンゴ〟; A$ RETURN **250 INPUT 〝ヤク〟; B$ RETURN**	シーケンシャルファイルに書かせるデータを用意します（キーボードからデータを入力できるように **input** を使用）。
300 PRINT ＃1, A$, B$ RETURN	データをファイルに書き込みます（**print＃**を使用）。
400 CLOSE ＃1 RETURN **450 OPEN 〝DATA〟 FOR OUTPUT AS ＃1 RETURN**	データを書き込んだファイルからデータを読み出すために一度ファイルを閉じてから，再び開きます。
500 INPUT ＃1, X$, Y$ RETURN	**input＃**によりファイルから，データを読み出します。
550 PRINT X$, Y$ RETURN	画面に表示させます。
600 CLOSE ＃1 RETURN	ファイルを閉じます。
700 END RETURN	

入力は終わりましたか？

まず，プログラムリストを見てみましょう。次のようにキー入力してください。

f・4 **RETURN**（または **list** **RETURN**）

間違いありませんね。もしあれば，直してください。それでは実行してみましょう。

f・5（または **run** **RETURN**）

すると画面に“エイタンゴ？”を表示してきました。何か英単語を入力してください。

```
(Sample) run
        エイタンゴ？ book [RETURN]
        ヤク？ ホン[RETURN]
        book ホン
        Ok
```

実行例をあげました。このような結果を得ることができましたか。

このプログラムでは1つの英単語しかファイルに書くことができません。次のようにプログラムを変えて下さい。

```
100 OPEN "DATA" FOR APPEND AS #1 [RETURN]
230 IF A$="end" THEN 400 [RETURN]
350 GOTO 200 [RETURN]
570 GOTO 500 [RETURN]
```

それでは実行してみましょう。

[f・5]（または　run [RETURN]）

ここでも，実行例を挙げておきましょう。

```
(Sample) run
        エイタンゴ？ time [RETURN]
        ヤク？ ジカン [RETURN]
        エイタンゴ？ end [RETURN]
        book ホン
        time ジカン
        Input past end in 500
```

行番号 100 の **OPEN** は，シーケンシャルファイルへデータを追加するために，モードを変更するためのものです。これで先程作ったファイルのあとにデータを追加していきます。

行番号 230 でキー入力の終りを検出します。

実行させたところ，**Input past end in 500** というエラーメッセージが表示されました。これは行番号 500 から 570 までが，終りのないループになっており，すべてのデータを読み終っても，行番号 500 の **input#** を実行しようとしたために，表示されたわけです。このエラーをなくすためには，ファイルの終りを検出して，その後 **INPUT#** を実行しないようにすればよいわけです。ファイルの終りを検出する関数とし

て **EOF**(**End Of File**の略)関数があります。

この関数を使って次のように変更してみましょう。

```
50 CLOSE
470 IF EOF(1) THEN 600
570 GOTO 470
```

行番号 50 の **CLOSE** は，エラーのためにクローズされなかったファイルを一度クローズするために入れておきます。

今度はどうでしょうか。実行させてみましょう。

[f・5] (または **run** [RETURN])

下の実行例のようになりましたか。

(Sample)
```
run
エイタンゴ? desk
ヤク? ツクエ
エイタンゴ? end
book ホン
time ジカン
desk ツクエ
Ok
```

以上で，プログラムが完成しました。プログラムを実行させるごとに，単語数は増えていきます。

8.6.4 ランダムアクセスファイルにおけるデータのロード・セーブ

ランダムアクセスファイル（以後、ランダムファイルと呼びます）は，どのようなときに使われるのでしょうか。住所録を例にとってランダムファイルの使い方を説明しましょう。

本棚から，住所録をとり出して，駒津さんの電話番号を調べ，住所録を本棚に戻すまでの手順を，ランダムファイルからのデータの読み出しと照らしあわせて説明していきます。

(1) 本棚から住所録を選んでとり出します。

(2) この住所録は，1ページに一人分のデータ（名前，住所，電話

番号，年令など）が規則正しく書き込まれています。

(3) 駒津さんのデータが書いてあるページを開き，そのページのコピーをとります。(このデータは， 23 ページにあるものとします。)

(4) コピーの中から，電話番号の書いてある部分の情報を，切り取ります。

(5) 本棚に住所録を戻します。

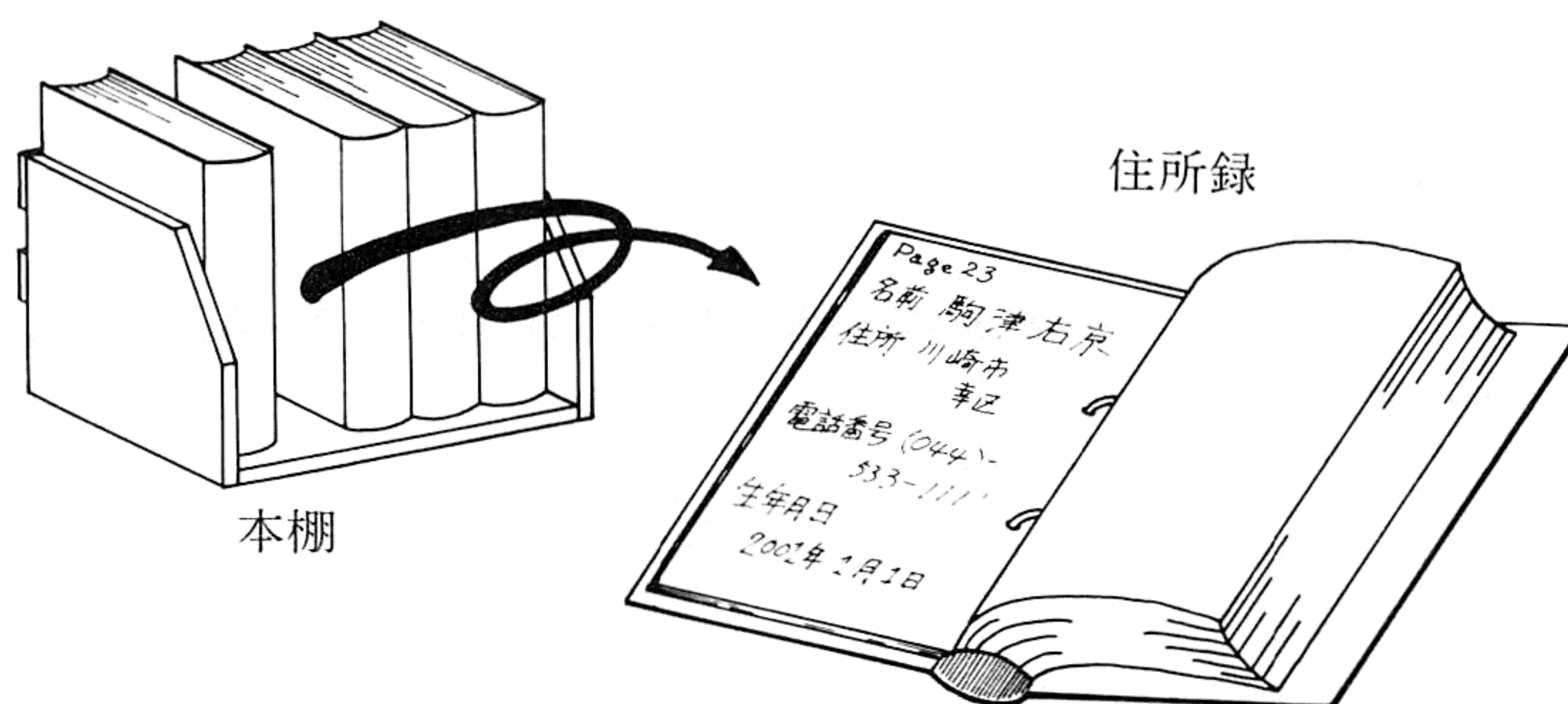



図８．７　電話番号を調べる

この例を実際にランダムファイルを扱う場合に当てはめてみると

本棚	：	フロッピィディスク
住所録	：	"jusho" という名まえのランダムファイル
1ページ	：	1レコード
1ページの中のデータの割り当て	：	**FIELD**
23 ページのコピーをとる	：	**GET**
コピーから切りとった電話番号	：	**TEL$**
本棚から住所録をとり出す	：	**OPEN**
本棚に住所録を戻す	：	**CLOSE**

のように対応します。

電話番号を調べる手順をプログラムにすると次のようになります。ランダムファイルにおける **データのロードの例** です。

```
20 OPEN "jusho" AS #1
30 FIELD #1, 20 AS NAMAE$, 30 AS JUSHO$, 25 AS TEL$, 10
AS AGE$
140 GET #1, 23
150 PRINT TEL$
200 CLOSE #1
```

また，住所録を作るプログラムは，次のようになります。これがランダムファイルにおける**データのセーブの例**です。

```
10 I=0
20 OPEN "jusho" AS #1
30 FIELD #1, 20 AS NAMAE$, 30 AS JUSHO$, 25 AS TEL$, 2
AS AGE$
40 LINE INPUT "ナマエ"; A$
50 IF A$="end" THEN GOTO 200
60 I=I+1
70 LINE INPUT "ジューショ"; B$
80 LINE INPUT "デンワバンゴウ"; C$
90 LINE INPUT "ネンレイ"; D$
100 LSET NAMAE$=A$
110 LSET JUSHO$=B$
120 RSET TEL$=C$
130 RSET AGE$=MKI$(VAL(D$))
140 PUT #1, I
200 CLOSE #1
```

セーブの例とロードの例を比較しながら手順を説明していきます。

(1)　**ファイルを開く（OPEN）（行番号20）**

ファイルのオープンの方法は，シーケンシャルファイルの場合とほとんど同じです。唯一のちがいは，**FOR INPUT**，**FOR OUTPUT**，**FOR APPEND** のオープンモードを指定しないことです。従ってロードの例もセーブの例も同じように **OPEN** すればよいのです。

Sample **OPEN "Jusho" AS #1**

これで "**Jusho**" というファイルに，ファイル番号1のランダムバッファが割り当てられます。

(2) フィールドの定義(FIELD)(行番号30)

ランダムファイルは，1レコードが256バイトに固定されます。この256バイトの中をどのように区切ってデータを納めるかを **FIELD** で定義します(これをフィールドの定義といいます)。

書　式　**FIELD #〈ファイル番号〉,〈フィールドの指定〉** ………

ロードの例，セーブの例では，同じようにフィールド定義を行っています。

Sample　**FIELD #1, 20 AS NAMAE$, 30 AS JUSHO$, 25 AS TEL$, 2 AS AGE$**

この意味は，ファイル番号1のランダムファイルのレコードの最初の20バイトを **NAMAE$**，次の30バイトを，**JUSHO$**，次の25バイトを **TEL$**，次の2バイトを **AGE$** の各変数に割り当て，残りの256－(20＋30＋25＋2)＝179バイトは使用しないということです。**FIELD**は，レコードがどのように使われ　かを定義するだけでデータをフロッピィディスクに書くこともしなければランダムバッファに書くこともしません。

注　意　1つのファイル番号に対する **FIELD** 定義が1行(256文字)に納まらないときは次のようにします。

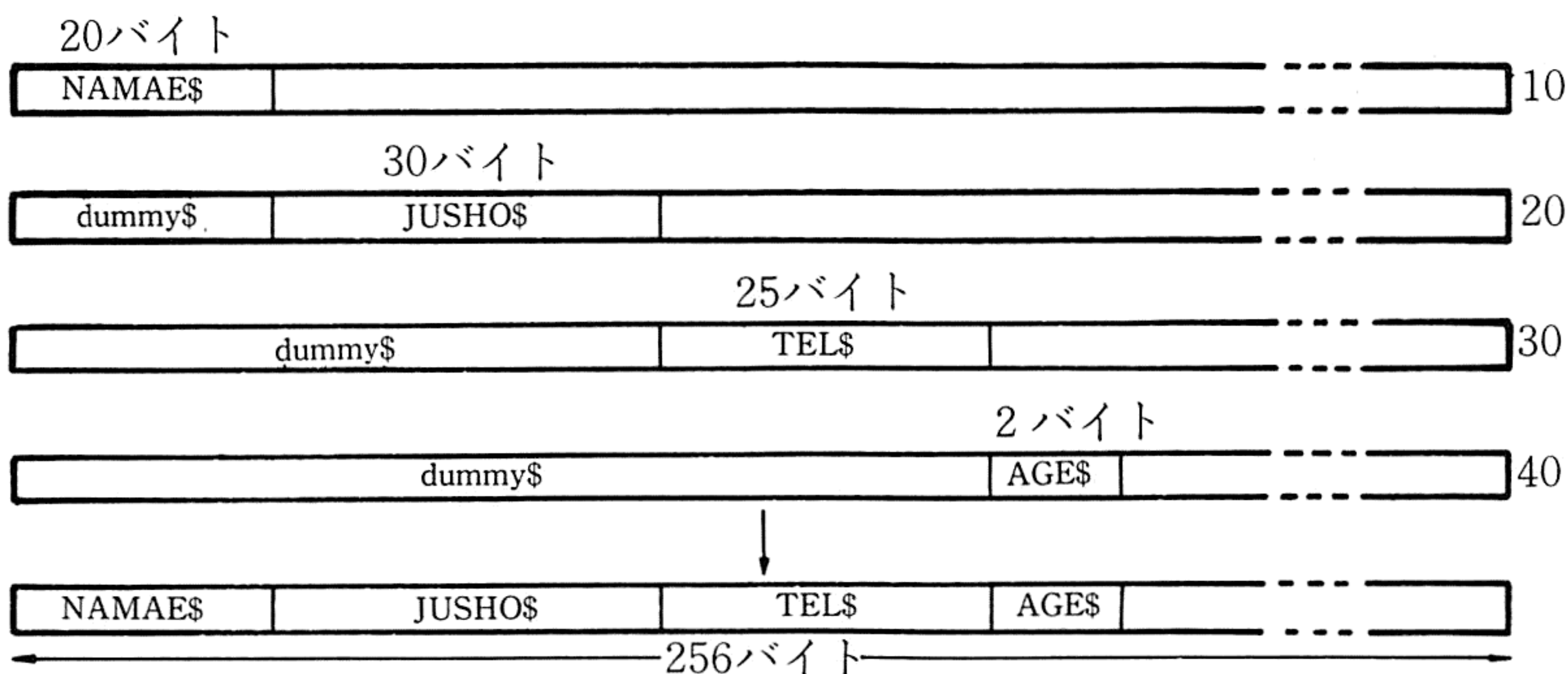
Sample

```
10 FIELD #1, 20 AS NAMAE$
20 FIELD #1, 20 AS DUMMY$, 30 AS JUSHO$
30 FIELD #1, 50 AS DUMMY$, 25 AS TEL$
40 FIELD #1, 75 AS DUMMY$, 2 AS AGE$
```

行番号10～40までを実行すると図8.8のようにフィールド定義されます。これは前ページのプログラムの行番号30と全く同じになります。

20バイト
NAMAE$ ……10
30バイト
dummy$ | JUSHO$ ……20
25バイト
dummy$ | TEL$ ……30
2バイト
dummy$ | AGE$ ……40
↓
NAMAE$ | JUSHO$ | TEL$ | AGE$
←256バイト→

図8.8 **FIELD** 定義

DUMMY$**を入れないと，**NAME$, **JUSHO$**, **TEL$**, **AGE$**はすべて256バイトのはじめに定義され（同じ位置に重ねて定義され）最終的には**AGE$** しか残りません。

行番号10～30のようにフィールド定義を行ったあと，次項のデータのセット(**LSET**, **RSET**)で，**DUMMY$**には，何もセットしなければよいのです。

(3) ランダムバッファへのデータの書き込み(LSET,RSET)(行番号100～130)

FIELDで定義したランダムバッファの各フィールドに，実際にディスクに書き込まれるべきデータを書き込むには，**LSET**, **RSET**の各ステートメントを使います。

書　式 **LSET <field定義に用いた文字変数>＝<文字式>**
RSET <field定義に用いた文字変数>＝<文字式>

LSET, **RSET** は，データのセーブのときだけ必要です。

Sample
```
FIELD ＃1, 20 AS NAMAE$, 30 AS JUSHO$, 25 AS TEL$,
2 AS AGE$

LSET NAMAE$=″コマツ ウキョウ″
LSET JUSHO$=″カワサキシ サイワイク″
RSET TEL$= ″044—533—1111″
RSET AGE$= MKI$(26)
```

ここには，文字式を定数で使いましたが，セーブの例のように変数で使うこともできます。ランダムバッファ ＃1 には次のようにデータが書き込まれます。

20バイト	30バイト	25バイト	2バイト	179バイト
コマツ ウキョウ△△…△	カワサキシ　サイワイク△△…△△	△△…△△044－533－1111	S_B	△△△△……………………△△△△

←256バイト→

図8．9　**LSET**と**RSET**

注　意 ランダムファイルは，データをすべて，文字型で扱わなければなりません。そのためには，数値変数は，文字変数に変換する必要があります。
したがって，数値を，ランダムファイルに書き込むときは，**MKI$**, **MKS$**, **MKD$**で文字型に変換してから書き込みま

す。読み込んだ時は，**CVI, CVS, CVD** で数値に変換します。

数値のデータ

3338

mki$
mks$
mkd$

ランダムファイル

文字型のデータ

3338

cvi
cvs
cvd

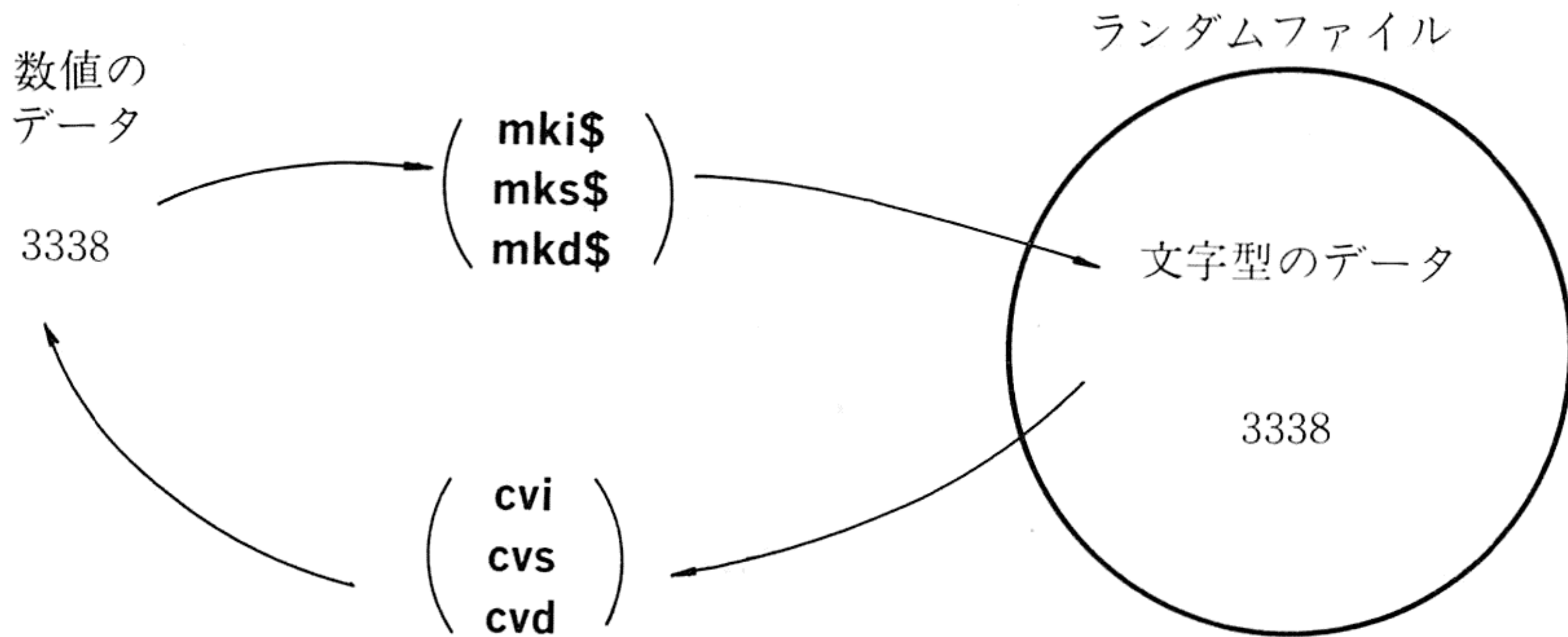



図 8.10　数値データの変換

LSET は **FIELD** 定義した文字変数の領域の左から文字式をつめます。**RSET** は右から文字式をつめます。

(注　意) **FIELD** で指定した文字変数に，データを割りあてるときは，必ず **LSET** か **RSET** で行います。

LET, INPUT, READ などを使ってはいけません。

(Sample) 悪い例

```
100 FIELD ♯1 100 AS A$, 100 AS B$
200 A$="DAME"
300 INPUT B$
400 PUT ♯1
```

(4)　ランダムバッファの内容を フロッピィディスク上の ランダムファイルに書く（PUT）（データのセーブ例の行番号140）

PUT はデータのセーブのときに使います。

(書　式) **PUT ♯〈ファイル番号〉〔,〈レコード番号〉〕**

データをセーブするときは，まずフィールド定義をし，**LSET, RSET** でランダムバッファにデータを書き込み，最後に **PUT** でフロッピィディスクにランダムバッファの内容を書き込みます。レコード番号で，ランダムバッファの内容を，ファイルの何番目のレコードとして書くかを指定します。

(Sample) **PUT #1, 3**

フロッピィディスク上のランダムファイル

レコード1	レコード2	レコード3	レコード4	----

PUT

ランダムバッファ

LSET, RSET

データ

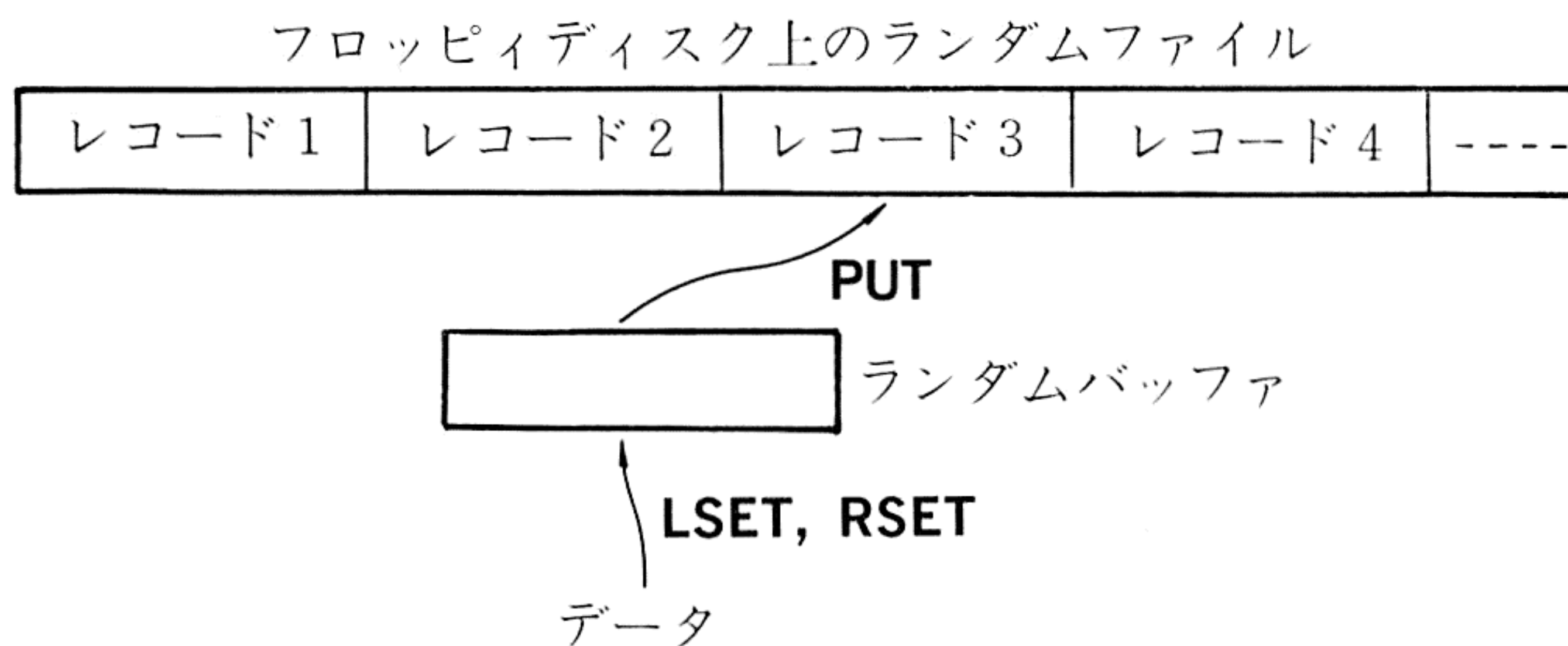



図8.11 **PUT**

〈レコード番号〉を省略した場合，直前に **GET** または **PUT** したレコードの次のレコードに書き込みます。

(5) フロッピィディスク上のランダムファイルからランダムバッファに読み込む(GET)（データのロード例の行番号140）

GET は，データのロードのときに使います。

(書　式) **GET #〈ファイル番号〉〔，〈レコード番号〉〕**

GETは，ディスクファイルから 1レコード文の データをランダムバッファに 読み込むと 同時に，そのデータを**FIELD**で 指定した 各 フィールドに 対応した 変数にそのデータを代 入します。

(Sample) **FIELD #1, 10 AS A$, 20 AS B$**
GET #1, 3

フロッピィディスク上のランダムファイル

レコード1	レコード2	レコード3	レコード4	------

GET

A$ B$

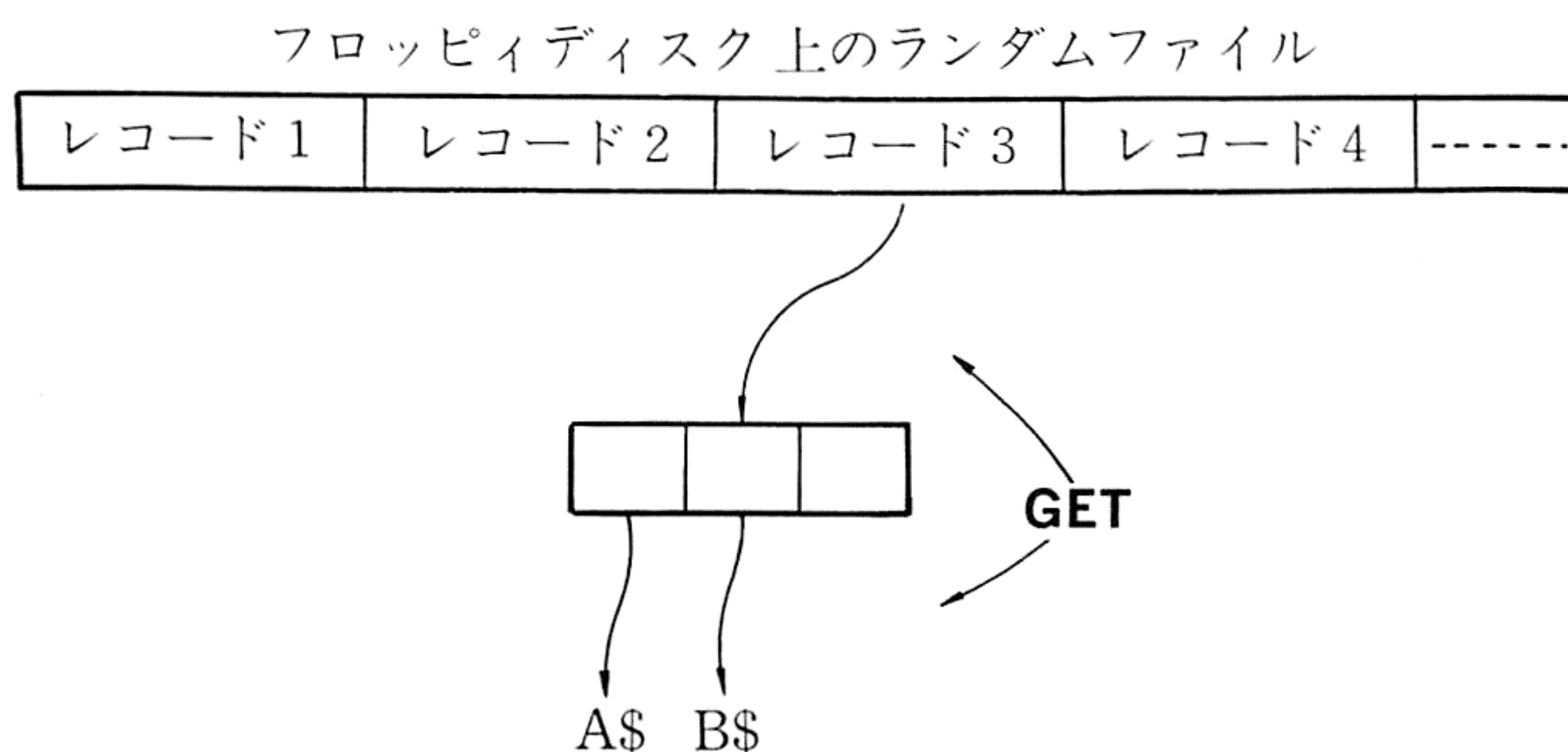



図8.12 **GET**

〈レコード番号〉を省略した場合，直前に **GET** または **PUT** したレコードの次のレコードから読み込みます。

(6) ファイルのクローズ(CLOSE)（行番号200)

シーケンシャルファイルと同様に **CLOSE** でファイルの使用終了宣言をします。

8.7 ディスクファイルの構造

この節では，**N80DISK-BASIC** におけるフロッピィディスクのファイル構造について説明します。この章は読みとばしてもかまいません。**DISK-BASIC** におけるフロッピィディスクのファイル構造については，ミニフロッピィディスクユニットなどに付いているユーザースマニュアルをお読みください。

8.7.1 フロッピィディスク

N80DISK-BASIC で使用できるフロッピィディスクは次の3種類です。

(1) 8インチ標準フロッピィディスク（両面倍密度用）

(2) 5インチミニフロッピィディスク（両面倍密度用）

(3) 5インチミニフロッピィディスク（片面倍密度用）

フロッピィディスクにはメモリと同じように番地がついています。フロッピィディスクの番地指定には**サーフェス番号**（両面倍密度のときのみ必要。0か1。)，**トラック番号**，**セクタ番号**を使用します。図 8.13 は8インチ両面倍密度用，図 8.14 は5インチ両面倍密度用，図 8.15 は5インチ片面倍密度用のフロッピィディスクの概略です。

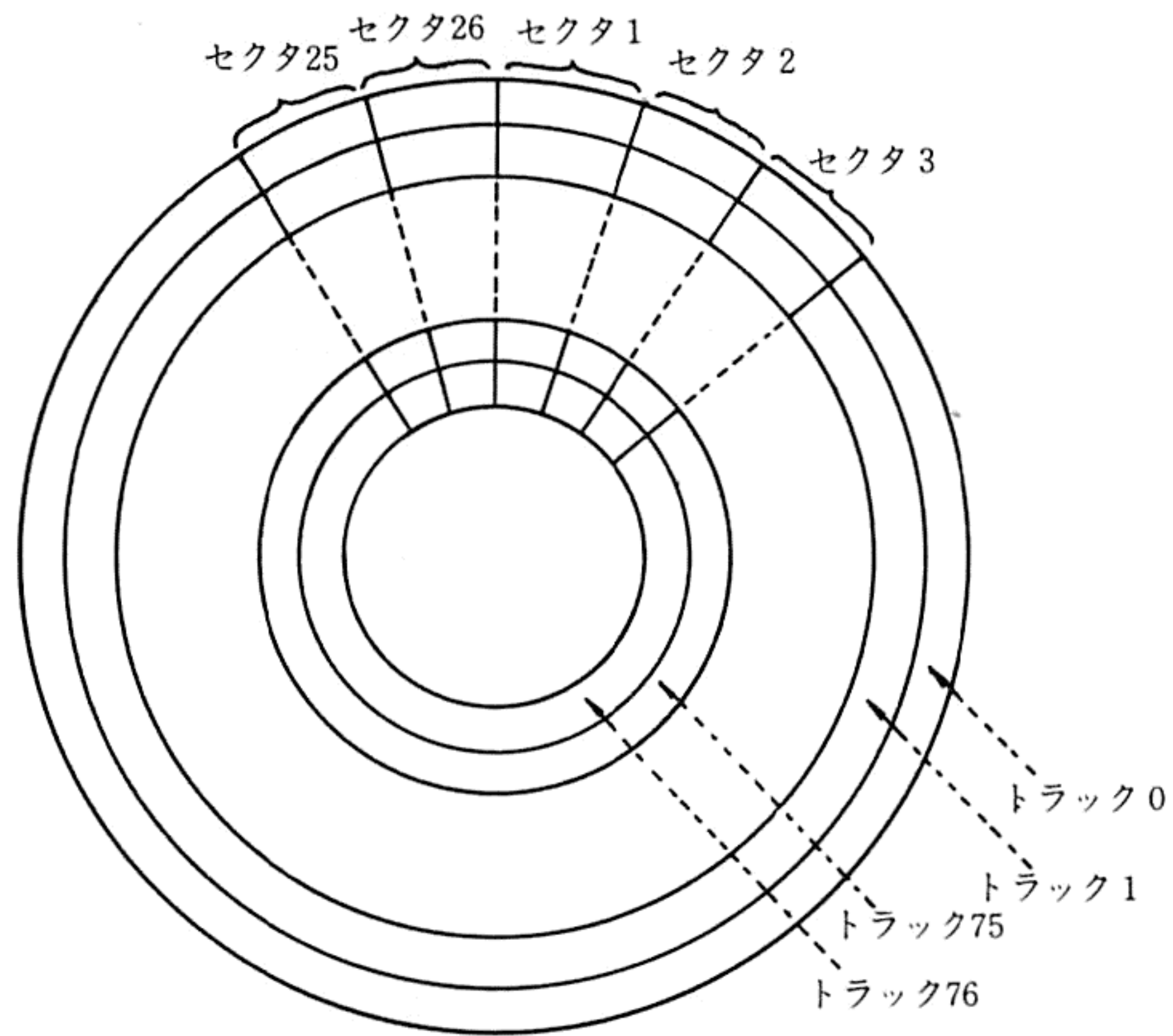



図８.13　８インチ両面倍密度用フロッピィディスク

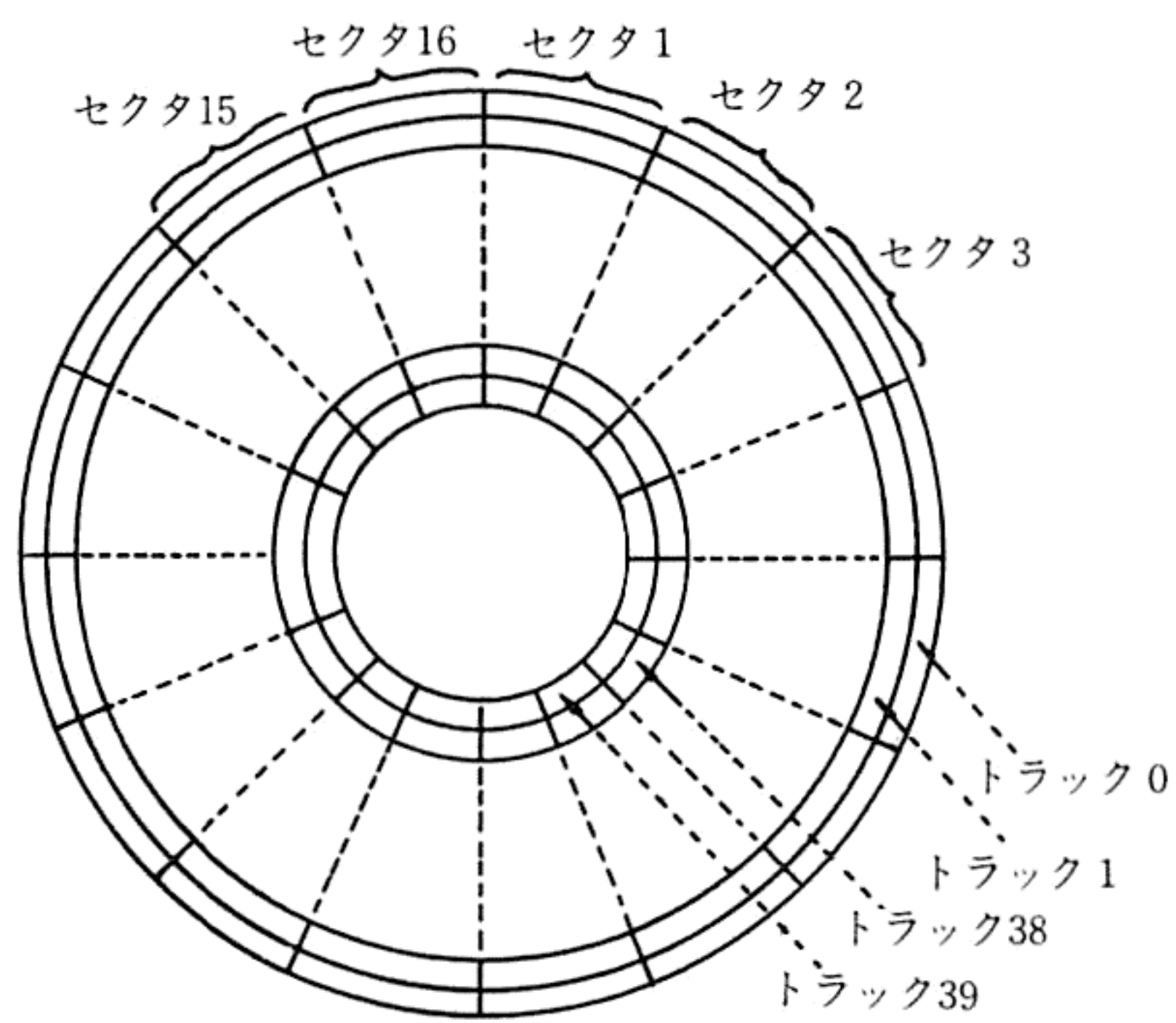



図８.14　５インチ両面倍密度用フロッピィディスク

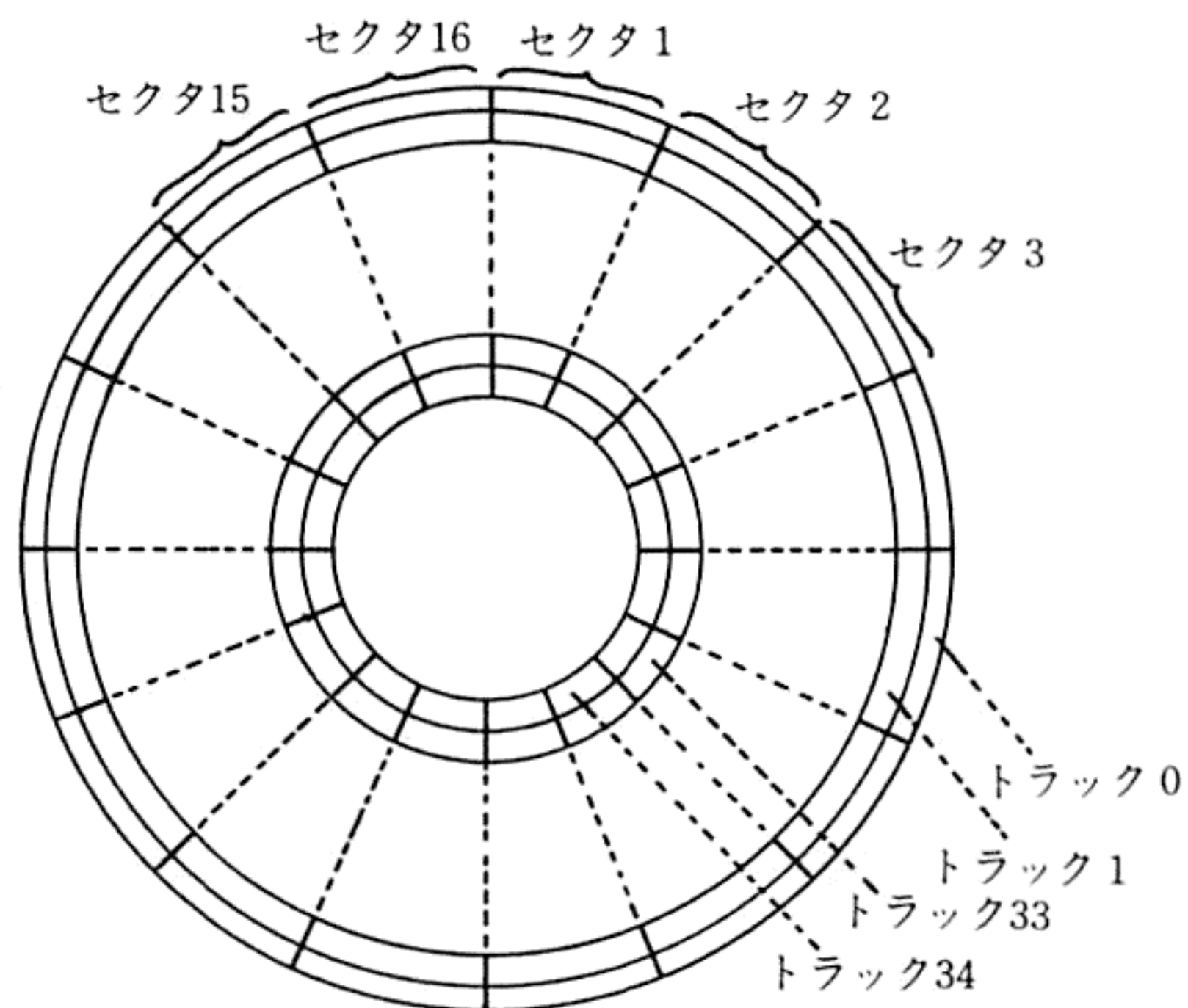



図８.15　５インチ片面倍密度用フロッピィディスク

図をみてもわかるように，トラック数，セクタ数はフロッピィディスクの種類によって異なります。これを表にまとめると次のようになります。

	サーフェス番号	トラック数	トラック番号	セクタ数	セクタ番号
8インチ 両面倍密度用	0か1	片面77(154)	0～76	26	1～26
5インチ 両面倍密度用	0か1	片面40(80)	0～39	16	1～16
5インチ 片面倍密度用	なし	35	0～34	16	1～16

Note **サーフェス番号0はおもて面ラベルの貼ってない方，サーフェス番号1はうら面です。3種類とも，1セクタは256バイトからなります。フロッピィディスクの場合，最小読み書き単位はセクタです。**

8.7.2 クラスタ，FAT，ディレクトリ

N80DISK-BASIC で，ディスクファイルを扱うときはすべてファイル名で行います。したがって，サーフェス番号，トラック番号，セクタ番号を使う必要はほとんどありません。ファイル名を，実際のディスクのサーフェス番号，トラック番号，セクタ番号に変換する仕事は **N80DISK-BASIC** が行なっています（図 8.16 参照）。

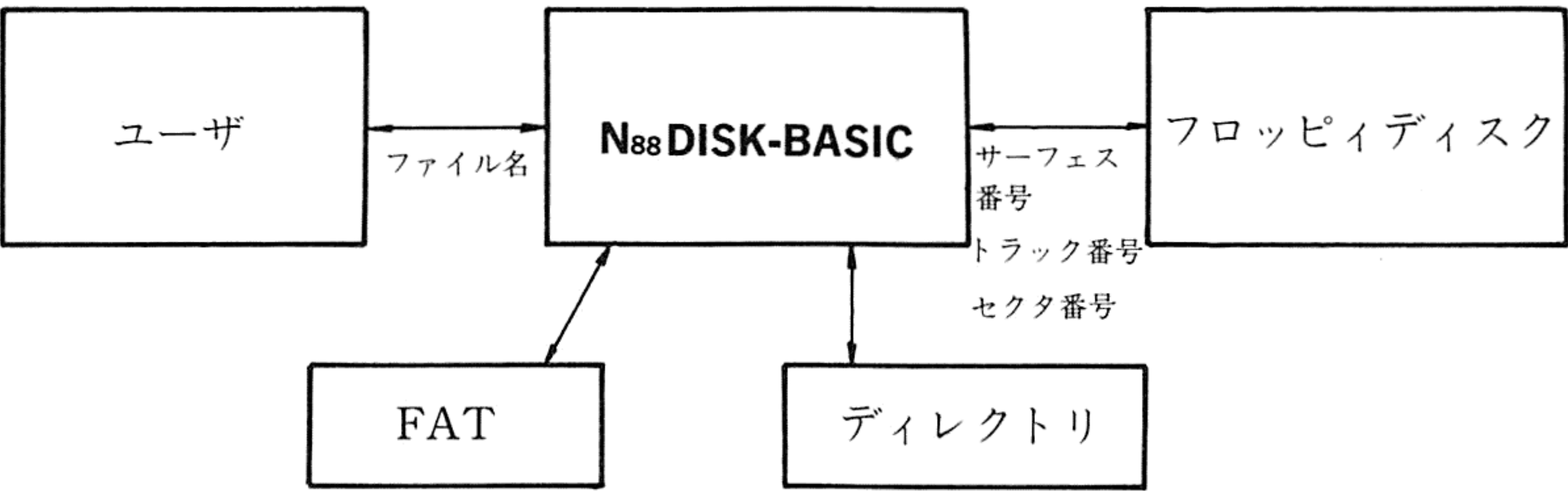



図8.16

ファイル名をフロッピィディスク上のアドレス（サーフェス番号，トラック番号，セクタ番号）に変換するために，**N80DISK-BASIC** では2つの表を管理しています。**ディレクトリ**と **FAT** がそれです。

ディレクトリは，そのフロッピィディスクに格納されているすべてのファイルの名前，属性，格納されている場所（の先頭）がかかれている表です。

FAT はフロッピィディスクの記憶場所の使用・未使用の状態を示す表です。

フロッピィディスクで読み書き可能な最小単位はセクタですが，**N$_{80}$DISK-BASIC** は 26 セクタ（5インチミニフロッピィディスクの場合は8セクタ）を1まとめにして，これを管理の最小単位としています。これを**クラスタ**と呼びます。クラスタとサーフェス，トラック，セクタの関係は次の表のようになっています。

表8．3　8インチ両面倍密度用フロッピィディスクの場合

クラスタ		サーフェス		トラック		セクタ	
10進	16進	10進	16進	10進	16進	10進	16進
0	0H	0	0H	0	0H	1～26	1H～1AH
1	1H	1	1H	0	0H	1～26	1H～1AH
2	2H	0	0H	1	1H	1～26	1H～1AH
3	3H	1	1H	1	1H	1～26	1H～1AH
4	4H	0	0H	2	2H	1～26	1H～1AH
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
152	98H	0	0H	76	4CH	1～26	1H～1AH
153	99H	1	1H	76	4CH	1～26	1H～1AH

（注　意）8インチ両面倍密度用フロッピィディスクはサーフェス0，トラック0は単密度になっているのでBASICで読み書きできません。

表8．4　5インチ両面倍密度用フロッピィディスクの場合

クラスタ		サーフェス		トラック		セクタ	
10進	16進	10進	16進	10進	16進	10進	16進
0	0H	0	0H	0	0H	1～8	1H～8H
1	1H	0	0H	0	0H	9～16	9H～10H
2	2H	1	1H	0	0H	1～8	1H～8H
3	3H	1	1H	0	0H	9～16	9H～10H
4	4H	0	0H	1	1H	1～8	1H～8H
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
158	9EH	1	1H	39	27H	1～8	1H～8H
159	9FH	1	1H	39	27H	9～16	9H～10H

表8.5　5インチ片面倍密度用フロッピィディスクの場合

クラスタ	トラック	セクタ	
		10進	16進
0	0	1〜8	1H〜8H
1	0	9〜16	9H〜10H
2	1	1〜8	1H〜8H
3	1	9〜16	9H〜10H
4	2	1〜8	1H〜8H
⋮	⋮	⋮	⋮
68	34	1〜8	1H〜8H
69	34	9〜16	9H〜10H

8.7.3 トラックの割り当て

(1)　8インチ両面倍密度用フロッピィディスク

表8.6　システムディスク

サーフェス	トラック		セクタ		内　　容
	10進	16進	10進	16進	
0	0	0H	すべて	すべて	未使用
	1	1H	1	1H	IPL
	1	1H	2〜26	2H〜1AH	DISK code
	2	2H	すべて	すべて	DISK code
	3〜34	3H〜22H	すべて	すべて	ユーザ領域
	35	23H	1〜22	1H〜16H	ディレクトリ
	35	23H	23	17H	ID
	35	23H	24〜26	18H〜1AH	FAT
	36〜76	24H〜4CH	すべて	すべて	ユーザ領域
1	0	0H	すべて	すべて	未使用
	1	1H	すべて	すべて	DISK code
	2〜76	2H〜4CH	すべて	すべて	ユーザ領域

表8.7　データディスク

サーフェス	トラック		セクタ	内　　容
	10進	16進		
0	0	0H	すべて	未使用
	1～34	1H～22H	すべて	ユーザ領域
	35	23H	すべて	システムディスクと同じ
	36～76	24H～4CH	すべて	ユーザ領域
1	0	0H	すべて	未使用
	1～76	1H～4CH	すべて	ユーザ領域

(2)　5インチ両面倍密度用フロッピィディスク

表8.8　システムディスク

サーフェス	トラック		セクタ		内　　容
	10進	16進	10進	16進	
0	0	0H	1	1H	IPL
	0	0H	2～16	2H～10H	}DISK code
	1	0H	すべて	すべて	
	2～39	2H～27H	すべて	すべて	ユーザ領域
1	0～1	0H～1H	すべて	すべて	DISK code
	2～17	2H～11H	すべて	すべて	ユーザ領域
	18	12H	1～12	1H～CH	ディレクトリ
	18	12H	13	DH	ID
	18	12H	14～16	EH～10H	FAT
	19～39	13H～27H	すべて	すべて	ユーザ領域

表8.9　データディスク

サーフェス	トラック		セクタ	内　　容
	10進	16進		
0	0～39	0H～27H	すべて	ユーザ領域
1	0～17	0H～11H	すべて	ユーザ領域
	18	12H	すべて	システムディスクと同じ
	19～39	13H～27H	すべて	ユーザ領域

(3)　5インチ片面倍密度用フロッピィディスク

表8.10　システムディスク

トラック		セクタ		内　　容
10進	16進	10進	16進	
0	0H	1	1H	IPL
0	0H	2～16	2H～10H	DISK code
1～3	1H～3H	すべて	すべて	
4～17	4H～11H	すべて	すべて	ユーザ領域
18	12H	1～12	1H～CH	ディレクトリ
18	12H	13	DH	ID
18	12H	14～16	EH～10H	FAT
19～34	13H～22H	すべて	すべて	ユーザ領域

表8.11　データディスク

トラック		セクタ	内　　容
10進	16進		
0～17	0H～11H	すべて	ユーザ領域
18	12H	すべて	システムディスクと同じ
19～34	13H～22H	すべて	ユーザ領域

8.7.4　ディレクトリ

各ディレクトリは 16 バイトからなります。各バイトは次のような内容を持っています。

表8.12　ディレクトリ

バイト		内　　容
10進	16進	
0～5	0H～5H	ファイル名
6～8	6H～8H	拡張子
9	9H	属性
10	AH	そのファイルが格納されている先頭のクラスタ番号
11～15	BH～FH	未使用

属性は次のようになっています。

表8.13　属性コード

値(16進数)	意　　味
0H	ASCII形式
10H	ASCII形式，書き込み禁止
40H	ASCII形式，リードアフターライト
80H	非アスキー形式
90H	非アスキー形武，書き込み禁止
C0H	非アスキー形式，リードアフターライト

ファイル名の先頭バイトの内容がFFHのとき，そのディレクトリは未使用であることを示します。0のとき，削除(**KILL**)されたファイルであることを示します。

8.7.5 FAT

FATは各クラスタの使用状態を示す表です。8.7.3のトラックの割りあての表をみてもらえばわかるように，**FAT**は3セクタにわたって書き込まれています。各セクタは全く同じ内容です。フロッピィディスクをアクセスしたときに，**FAT**の内容(3セクタ)がメモリにコピーされます。それぞれのセクタの0～153(8インチ両面倍密度の場合)バイトがクラスタ0～153にそれぞれ対応しています。すなわち各バイトの持っている値が，各クラスタの使用状態を示しているのです。

表8.14　FAT

バイトのデータ(16進)	クラスタの使用状態
8インチ両面の時 0～99 5インチ両面の時 0～9F 5インチ片面の時 0～45	使用中。連続したクラスタの一部であり，後続するクラスタを持つ。値が，後続するクラスタの番号を示している。
8インチ両面の時 C1～DA 5インチの時 C1～C8	使用中。連続したクラスタの最後のクラスタであり，下5(5インチの時は下4)ビットの内容が，そのクラスタで実際に使われているセクタの数を表わす。
FE	予約ずみのクラスタで，ファイルとして使うことはできない。(DISK code, IPL, ディレクトリ，FAT自身を含むクラスタがこれである。)
FF	未使用。

なお，表の説明で「連続したクラスタ」といいましたが，これは物理的に連続したクラスタを意味しません。**N80DISK-BASIC** が管理するファイルは一般に物理的にとびとびのクラスタをつないで構成されているのです。

8.7.6 ID

ID には1セクタ（256 バイト）が割り当てられています。各バイトは次の様な意味を持っています。

属性	ファイル数	**BASIC**テキスト
1バイト	1バイト	254バイト

図8.17　IDの構造

最初の1バイトは，そのディスクの属性を示しています。値とその意味はディレクトリ部のものと同じです。

通常のシステムディスクでは〈ファイル数〉として 255(FFH) が指定されていて，このようなときは起動時に〝**How many files?**″とたずねてきます。この〈ファイル数〉に0～15を指定すると，**N80DISK-BASIC**をスタートさせる時に，〝**How many files?**″を表示せずに，自動的にその値に設定されます。〈ファイル数〉の設定が終ると(〈ファイル数〉をFFHにして，〝**How many files?**″に対してキーボードから入力を行った場合でも)，次の〈**BASIC**テキスト〉の内容が自動的に実行されます。 つまり，**N80DISK-BASIC**をスタートさせる場合，システムディスクの**ID**に書かれている〈**BASIC**テキスト〉の内容は常に実行されます。

IDは，〈属性〉が0，〈ファイル数〉が255(FFH)，〈BASICテキスト〉がすべて0か32(0H か20H)が通常の状態です。

(Sample) ID セクタ

00	05	72　75　6E　22　41　22　………　00

(r)(u)(n)(″)(A)(″)

この例のように **ID** セクタの内容が書かれている場合，動的に5個のファイルバッファが用意され，“A”という名前のプログラムが走り出します。

ID を書き込む場合は，システムディスクに入っている「**setinfN80**」というプログラムを使ってください。

ID の〈属性〉は **SET** によって書き変えます。

8．8 ディスク関数とDSKO$

ここでは，下にあげたフロッピィディスクに対する関数と，**DSKO$** ステートメントについて説明します。

EOF	シーケンシャルファイルの終わりを検出。
MKI$	整数を２バイトの文字列へ変換。
MKS$	単精度実数を４バイトの文字列へ変換。
MKD$	倍精度実数を８バイトの文字列へ変換。
CVI	２バイトの文字列を整数型の数値に変換。
CVS	４バイトの文字列を単精度実数型の数値に変換。
CVD	８バイトの文字列を倍精度実数型の数値に変換。
LOF	ディスクファイルの大きさをセクタ数で表す。
DSKF	フロッピィディスクに関する情報を示す。
INPUT$	指定した長さの文字列をファイルから読み込む。
LOC	ランダムファイルの場合は次のレコード番号。シーケンシャルファイルの場合は今まで読み書きしたセクタ数。
ATTR$	フロッピィディスクまたはファイルの属性。
FPOS	ファイルがある物理セクタ番号。
DSKI$	指定したセクタから文字列を読み込む。
DSKO$	指定したセクタに文字列を書き込む。

8.8.1 EOF

EOF はシーケンシャルファイルが終わりかどうかを検出する関数です。ファイルからデータを繰り返して入力するような場合に，入力したデータが最後のデータかどうかを調べるのに **EOF** 関数を用います。

(書　式) **EOF** (〈**ファイル番号**〉)

(Sample) **IF EOF (1) THEN GOTO 100**

8.8.2 MKI$, MKS$, MKD$

ランダムファイルに数値データを書き込む場合，文字型に変換して書き込みます。その変数を行うのが **MKI$**，**MKS$**，**MKD$** です。

MKI$** は整数型の数値を 2 バイトの文字列に，**MKS$ は単精度実数型の数値を 4 バイトの文字列に，また **MKD$** は倍精度実数型の数値を 8 バイトの文字列に変換します。

(書　式) **MKI$** (〈**整数表記**〉)

MKS$ (〈**単精度実数表記**〉)

MKD$ (〈**倍精度実数表記**〉)

(Sample) **LSET I$ = MKI$(X%)**

8.8.3 CVI, CVS, CVD

数値型から文字型に変換されてファイルに書き込まれているデータを読み込んだ場合に，文字型からもとの数値型に変換する必要があります。

その変換を行うのが **CVI, CVS, CVD** です。

CVI は 2 バイトの文字列を整数型の数値に，**CVS** は 4 バイトの文字列を単精度実数型の数値に，**CVD** は 8 バイトの文字列を倍精度実数型の数値にそれぞれ変換します。

(書　式) **CVI** (〈**2 バイトの文字例**〉)

CVS (〈**4 バイトの文字列**〉)

CVD (〈**8 バイトの文字列**〉)

(Sample) **A = CVS (X$)**

8.8.4 LOF

LOF は，フロッピィディスクファイルの大きさをセクタ数で返す関数で，フロッピィディスクファイルがランダムファイルの場合はそれまでに書き込まれた最大のレコード番号を返します。ですからランダムファイルにおいて新しく書き込むデータのレコード番号を求めるような場合に**LOF**関数を利用できます。

(書　式) **LOF (〈ファイル番号〉)**

(Sample) **REC = LOF (1) + 1**

8.8.5 DSKF

DSKF はディスクファイルの構造に関する情報を返す関数です。

(書　式) **DSKF (〈ドライブ番号〉〔,〈機能〉〕)**

ここで〈機能〉は 0～10 の整数で指定し，それによって **DSKF** は次のような機能を持っています。

0 ：片面あたりの最大トラック番号
1 ：1トラックあたりのセクタ数
2 ：片面の場合は0，両面の場合は1
3 ：1トラックあたりのクラスタ数
4 ：クラスタの総数
5 ：ディレクトリが格納されているトラック番号
6 ：1クラスタあたりのセクタ数
7 ：最初の **FAT** が格納されているセクタ番号
8 ：最後の **FAT** が格納されているセクタ番号
9 ：**FAT** の個数
10 ：**ID** が格納されているセクタ番号

〈機能〉を省略した場合は，そのフロッピィディスクの残りの容量をクラスタ単位で返します。

(Sample) **CRST = DSKF(1,4)**

8.8.6 INPUT$

INPUT$は，シーケンシャルファイルから指定した長さの文字列を読み込んで返す関数です。

(書　式) **INPUT$** (〈**文字列の長さ**〉〔，〈**#ファイル番号**〉〕)

ファイル番号を省略した場合は，キーボードから入力した文字列を読み込みます。詳しくはN$_{80}$-BASICリファレンスマニュアルを参照してください。

(Sample) **X$ =INPUT$ (5,#1)**

この例では，#1のシーケンシャルファイルから文字列を5文字読み込んで**X$**に代入します。

8.8.7 LOC

LOC関数はランダムファイルとシーケンシャルファイルとで機能が異なります。

(i) ランダムファイルの場合

GETまたは**PUT**が最後に実行されたときに使用されたレコード番号を返します。

(ii) シーケンシャルファイルの場合

OPENを実行した時からの読み書きされたレコード数を返します。

(書　式) **LOC** (〈**ファイル番号**〉)

(Sample) **j=LOC (1)**

8.8.8 ATTR$

ATTR$は，フロッピィディスクあるいはファイルの属性を返します。「属性」については8.5.6を参照して下さい。

(i) フロッピィディスクの属性

(書　式) **ATTR$** (〈**ドライブ番号**〉)

(ii) ファイルの属性

(書　式) **ATTR$** (〈**#ファイル番号**〉)

ATTR$ (〈**ファイル名**〉)

それぞれ，指定したフロッピィディスク，オープンされているファイルの現在の属性，または，ファイル名で指定したファイルの固定属性（ディレクトリィに記録されている属性）を表わす文字列を返します。

書き込み禁止　　　　　　：“△△ P”

READ AFTER WRITE ：“R △△”

属性なし　　　　　　　　：“△△△”

(Sample) **PRINT ATTR$ (“PROG. BAS”)**

この例では，ファイル“**PROG. BAS**”の属性を求めます。

8.8.9 FPOS

ファイル番号によって指定されたファイルが最後に読み書きした位置をセクタの通し番号で示します。

セクタの通し番号は次のように割りあてます。

(i) 片面のフロッピィディスクの場合

トラック0のセクタ1に通し番号0を割当て，以下順に1きざみで割りあてていきます。例えば，トラック1のセクタ2は通し番号 17 になります。

(ii) 両面のフロッピィディスクの場合

サーフェス0のトラック0のセクタ1を通し番号0として，順にそのトラック内のセクタに番号を割りあてます。

次に，サーフェス1，トラック0のセクタに番号を続けて割りあてます。以下同様に，1トラックごとにサーフェスを換えて順に通し番号を割りあてます。例えば，8インチ標準フロッピィディスクの場合は次のように割りあてていきます。

サーフェス	0	0	0…0	0	1	1	1…1	1	0	0	0……
トラック	0	0	0…0	0	0	0	0…0	0	1	1	1……
セクタ	1	2	3…25	26	1	2	3…25	26	1	2	3……
通し番号	0	1	2…24	25	26	27	28…50	51	52	53	54……

(書　式) **FPOS （〈ファイル番号〉）**

(Sample) **PRINT FPOS (1)**

8.8.10 DSKI$とDSKO$

ここでは，関数 **DSKI$** とステートメント **DSKO$** の説明をします。**DSKO$** は非常に危険なステートメントです。したがって，このステートメントを使用するには十分注意をして下さい。このステートメントを実行する前に，使用するフロッピィディスクのバックアップを作っておくとよいでしょう。

8.8.10.1 DSKI$

フロッピィディスクの内容を読み込む場合，これまで説明してきた命令や関数では ファイル名か ファイル番号 で その内容を指定しましたが，**DSKI$** は物理アドレス（ドライブ番号，サーフェス番号，トラック番号，セクタ番号）で指定した内容を読み込む関数です。

(i) 片面のフロッピィディスクを使用する場合

(書　式) **DSKI$ （〈ドライブ番号〉，〈トラック番号〉，〈セクタ番号〉）**

(Sample) **D$＝DSKI$ (2, 34, 16)**

この例では，ドライブ2，トラック 34，セクタ 16 の1セクタの内容を読み込みます。

(ii) 両面のフロッピィディスクを使用する場合

(書　式) **DSKI$ （〈ドライブ番号〉，〈サーフェス番号〉，〈トラック番号〉，〈セクタ番号〉）**

(Sample) **D$ ＝DSKI$ (2, 1, 39, 16)**

この例では，ドライブ2，サーフェス1，トラック 39，セクタ16 の 1セクタの内容を読み込みます。

DSKI$ は1セクタのデータ（256 バイト）を文字列として返します。

ところが文字変数の最大の長さは 255 文字なので，上の(i)，(ii)の例では1セクタの最後の1文字が失われます。

DSKI$は＃0という特別なファイル番号のバッファに，指定したセクタの内容を読み込みますので，256バイト全部を読み込むには，次のようにします。

Sample
```
10 FIELD #0, 128 AS A$,128 AS B$
20 D$=DSKI$ (2, 1, 39, 16)
```

この例では，ドライブ2，サーフェス1，トラック 39，セクタ 16 の1セクタの前半の 128 バイトを**A$**に，後半の 128 バイトを**B$**に読み込みます。またそのセクタの前から 255 バイトは**D$**に代入されます。

このように **DSKI$** は，ランダムファイルと似たような使い方をします。

8.8.10.2 DSKO$

DSKO$** は **DSKI$ と同様に，ファイル番号＃0のバッファを使用します。まず，**FIELD ＃0** によって指定したバッファにデータを書き込みまみます。そして **DSKO$** によって，そのバッファの内容を指定したセクタに書き込みます。（**FIELD ＃0** の指定は「8.6.4」の中の**FIELD** の説明を参照してください。）

(i) 片面のフロッピィディスクを使用する場合

書　式 **DSKO$ 〈ドライブ番号〉，〈トラック番号〉，〈セクタ番号〉**

(ii) 両面のフロッピィディスクを使用する場合

書　式 **DSKO$ 〈ドライブ番号〉，〈サーフェス番号〉，〈トラック番号〉，〈セクタ番号〉**

Sample
```
DSKO$ 2, 1, 39, 16
```

この例では，ランダムファイルの場合と同様にしてセットした＃0のバッファの内容を，ドライブ2，サーフェス1，トラック39，セクタ16に書き込みます。

8.9　ユーティリティプログラムの使い方

この節では，システムディスクに入っているユーティリティプログラムの使い方を説明します。

8.9.1　format N80
（フロッピィディスクのフォーマッティング）

フロッピィディスクを初期化することを「フォーマッティングする。」といいます。フォーマッティングのしかたには2段階あります。ひとつは「物理フォーマット」もうひとつは「システムフォーマット」です。

(1)　物理フォーマット

「8.7.1 フロッピィディスク」で述べた，サーフェス，トラック，セクタといったアドレスをフロッピィディスクにつけることを物理フォーマットといいます。

市販されているミニフロッピィディスクには，このようなアドレスはつけられていません。また，8インチ標準フロッピィディスクも，市販のものは，**IBM フォーマット**とよばれる物理フォーマットがされているため，**N80 DISK-BASIC** 用に NEC が供給している **PC-8886** とは異なる物理フォーマットとなっています。ただし，**format N80**では，8インチ標準フロッピィディスクの物理フォーマットはできません。8インチ標準フロッピィディスクの物理フォーマットは，次節で説明する **phyfrm N80** で行います。

物理フォーマットを行うと，**N80DISK-BASIC** の **DSKI$**，**DSKO$** で読み書きができるようになります。

(2)　システムフォーマット

システムフォーマットを行ったフロッピィディスクは，**N80-DISK-BASIC** で使えるようになります。システムフォーマットでは，**FAT**，ディレクトリ，**ID** の初期化を行います。

format N80 は，ミニフロッピィディスクの物理フォーマットと，ミニフロッピィディスク，8インチ標準フロッピィディスクのシステムフォーマットを行うためのユーティリティプログラムです。

使用手順は次に示すとおりです。

(1) **N80DISK-BASIC** をスタートさせます。

(2) **format N80** をロードします。

load "<ドライブ番号> formatN80" RETURN

<ドライブ番号>は，システムディスクが入っているドライブ番号です。通常は 1：ですから省略できます。

(3) プログラムを実行させます。

f・5 または **run RETURN**

(4) 次のメッセージが表示されます。

Mount a new disk on even drive#.
Format drive#?

(5) 偶数のドライブ番号のついているドライブに新しいフロッピィディスクをセットします。(他のドライブには，フロッピィディスクを入れないでください。システムディスクが入っているドライブはプログラムのロードが終った事を確認したら，ふたを開けておきます。)

(6) 新しいフロッピィディスクがセットできたら，セットしたドライブのドライブ番号を入力します。ドライブ番号が 2 の場合は次のようになります。

2 RETURN

(7) 指定したドライブがミニフロッピィディスクユニットの場合は，次のメッセージが表示されます。

Do you need physical formatting(y/n)?

新しいミニフロッピィディスクで物理フォーマットを行う必要がある場合は，

y RETURN

一度物理フォーマットしたもので，その必要がない場合は，

n RETURN

を入力します。

“y” を入力すると，この場合ですと

Formatting a disk 2 RETURN

と表示して，物理フォーマットを行います。

“n” を入力すると物理フォーマットは行いません。

(8) 次にシステムディスクを作るかどうかたずねてきます。

Create system disk(y/n)?

システムディスクを作りたい場合は，

y RETURN

その必要がない場合は，

n RETURN

を入力します。

(9) システムディスクを作る，作らないにかかわらず，

Working

と表示して，システムフォーマッティングを行います。

(10) システムディスクを作らない場合は，

Complete.

と表示して実行を終了します。

システムディスクを作る場合は，次のメッセージが表示されます。

System disk on drive＃?

(11) システムディスクをセットしたら，そのドライブのドライブ番号を入力します。ドライブ番号が 1 の場合は次のようになります。

1 RETURN

(12) システムディスクをセットしたドライブを指定すると

Sure(y/n)?

と確認を求めてきます。もう一度確認して正しければ，

y RETURN

もしまちがっていれば，

n **[RETURN]**

を入力します。

"**n**"を入力すると，⑽にもどります。"**y**"を入力すると，

Copying system

と表示して，DISK code をコピーし，無事終了すると，

Complete.

と表示します。

ここで，新しいフロッピィディスクをシステムディスクとしてフォーマッティングした場合は，そのフロッピィディスクで **N$_{80}$DISK-BASIC** をスタートさせることができます。

(注　意) フォーマッティングを行うと，たとえ書き込み禁示を宣言してあってもそのフロッピィディスクに書き込まれていたファイルはすべて消去されます。ですから，うっかり重要なフロッピィディスクをフォーマッティングしたりすることのないように注意してください。

フロッピィディスクを，機械的に書き込み禁止にする方法があります。ミニフロッピィディスクの場合は，図 8.18 のようにライトノッチにライトプロテクトシールを貼ると，書き込み禁止になります。

ライトノッチ

ライトプロテクトシールを 貼ると，書き込み禁止となる。

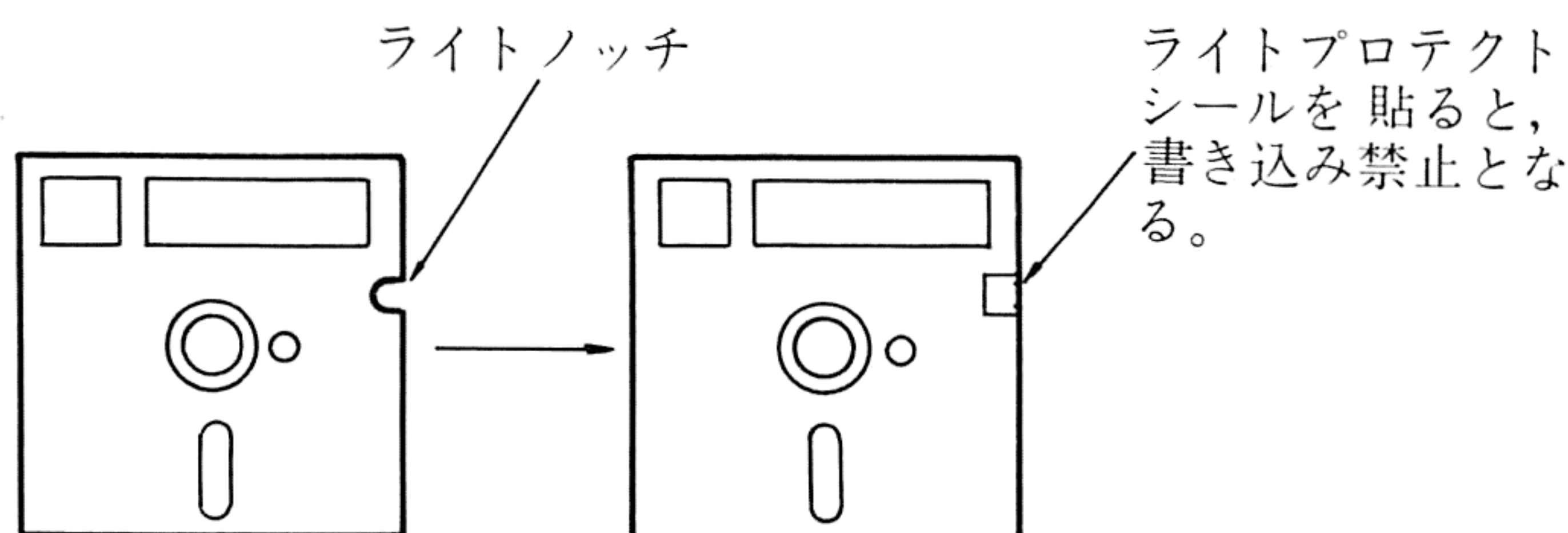



図 8.18　ミニフロッピィディスクのライトプロテクト

8 インチ標準フロッピィディスクの場合，通常は，機械的に書き込み禁止にすることはできません。システムディスク(**PC-8087**)は例外です。ライトシールを貼っていない状態では，書き込み禁止となっており，ライトシールを貼ると，書き込みできるようになります。

ライトプロテクトノッチ
この状態では書き込み禁止

ライトシールを
貼ると書き込み
できる。

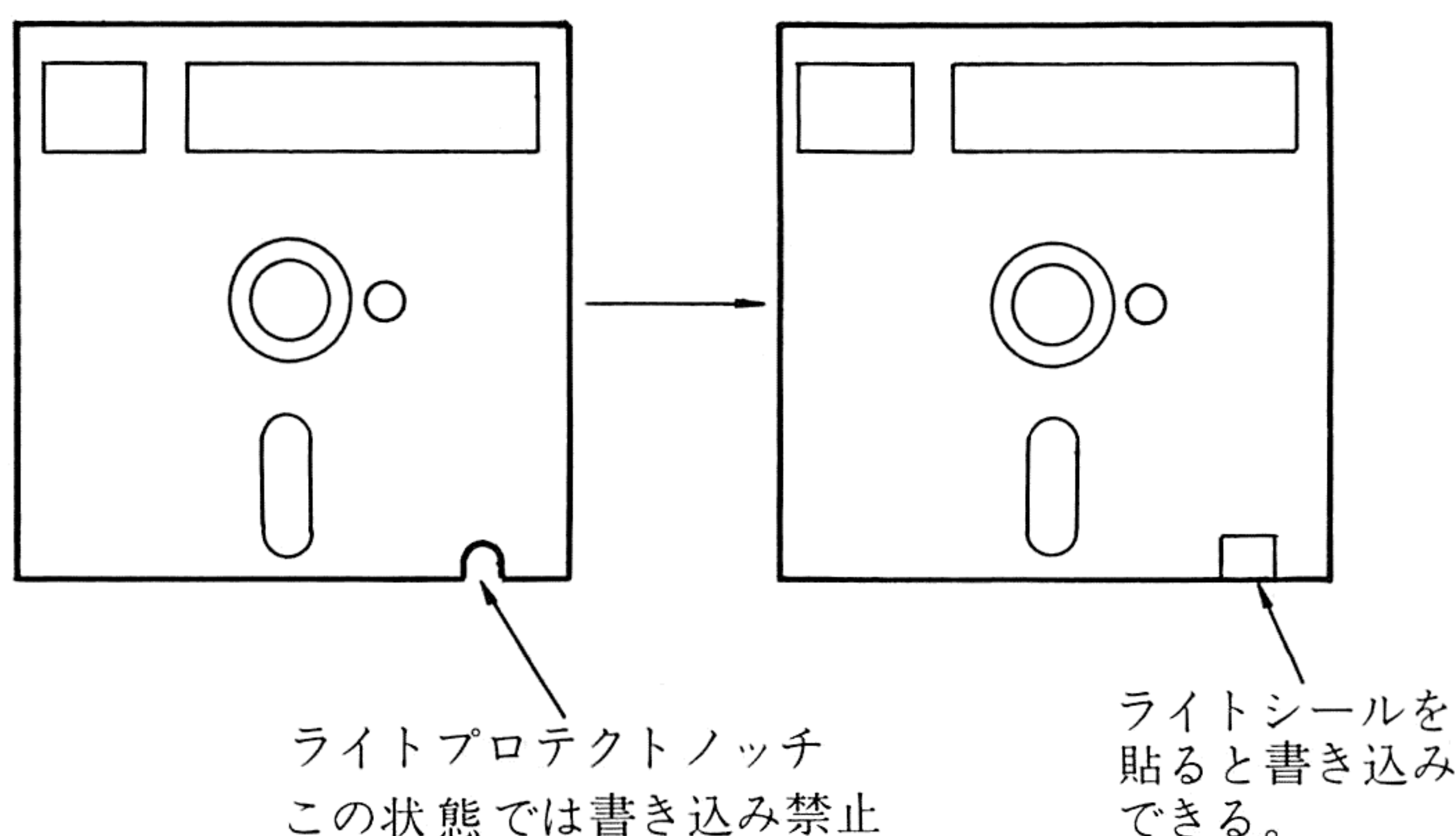



図8.19 8インチ標準フロッピィディスク(PC-8087)のライトプロテクト

(備　考) ミニフロッピィディスクのライトプロテクトシールと，8インチ標準フロッピィディスクのライトシールは，同じものです。機能がちがうので，ちがう呼び方をします。

8.9.2 phyfrmN80(8インチ標準フロッピィディスクの物理フォーマッティング)

phyfrmN80は，8インチ標準フロッピィディスクの物理フォーマットを行なうためのユーティリティプログラムです。 **N80DISK-BASIC** 用にNEC が供給している **PC-8886** は，物理フォーマットされていますから， **phyfrmN80** を使う必要はありません。

使用手順は，次に示すとおりです。

(1) **N80DISK-BASIC** をスタートさせます。

(2) **phyfrmN80** をロードします。

load "＜ドライブ番号＞ phyfrmN80" **[RETURN]**

＜ドライブ番号＞は，システムディスクが入っているドライブ番号です。通常は1ですから省略できます。

(3) プログラムを実行させます。

[f・5] または **run** **[RETURN]**

(4) 次のメッセージが表示されます。

Physical formattor for DMA based drive

Please remove the system disk to prevent from illegal operations

Please wait for a few seconds

「**DMA based drive** 」とは **PC-8881** と **PC-8882** のドライブのことです。

(5) 数秒たつと次のメッセージが表示されます。

Mount an unformatted disk on drive
Format drive#? 2
Sure(y/n)? y

Format drive # ? に対しては，物理フォーマットをする8インチ標準フロッピィディスクの入っているドライブ番号を入力します。

Sure (y/n) ? に対しては，もう一度そのフロッピィディスクを物理フォーマットしてもよいか確かめた上で，小文字の **y** を入力してください。**y** 以外の文字を入力すると，もう一度(5)を操り返します。

ミニフロッピィディスクの入っているドライブ番号を指示した時は，次のメッセージが表示されます。

Sorry, Can't format.

You can format a disk for PC-8031 by formatN80/backupN80.

従って，「**formatN80** か，**backupN80** で物理フォーマットしてください。」というわけです。

(6) 次のメッセージが表示され，トラックごとに，ピリオドを表示しながら，物理フォーマットをしていきます。

Formatting a disk 2
. . . . .
. . . . .

(7) 物理フォーマットが終わるとチェックをします。　この途中でDisk I/O error が出たときは，そのフロッピィディスクは，物理フォーマットできないということで，残念ながら，捨てるしかありませ

ん。Complete. というメッセージが出れば，物理フォーマットは終わりです。

```
Checking the formatted disk
................................
................................
Complete.
```

8.9.3 backupN80(フロッピィディスクのフォーマッティングとバックアップ)

フロッピィディスクにセーブされたファイルは壊れることがあります。例えば，まちがって必要なファイルを **KILL** で消してしまったり，**DSKO$** でファイルの内容を壊したり，また，ノイズなどハードウェアのトラブルなどで壊れる可能性もあります。

そこで，重要な情報が記録されているフロッピィディスクは，定期的に複写を作っておいて，もしファイルが壊れても，その複写ファイルを用いて，その被害を最小限にすることが必要です。このような複写ファイルを作ることを「バックアップファイルを作る」と言います。順を追って「バックアップファイルの作り方」を説明します。

○バックアップファイルの作り方

システムディスクには，**"backupN80"** というバックアップファイル作成プログラムが入っています。このプログラムでは，8インチ標準フロッピィディスクは8インチ標準フロッピィディスクに，ミニフロッピィディスクはミニフロッピィディスクにバックアップします。ミニフロッピィディスクのバックアップファイルを8インチ標準フロッピィディスクに作ったり，あるいはその逆を行うことはできません。

(1) まず，**N$_{80}$DISK-BASIC** をスタートさせ，バックアップファイル作成プログラムをロードします。

load "＜ドライブ番号＞ backupN80" RETURN

(2) このプログラムを実行します。

f・5 または **run RETURN**

(3) すると画面に次のメッセージが現われます。

Mount master disk on drive.
Master drive#?

バックアップファイルを作りたいフロッピィディスク（マスターファイル）をドライブにセットし，そのドライブのドライブ番号を入力します。ドライブ番号1のドライブにセットした場合は，

1 RETURN

と入力します。

(4) 次に新しいフロッピィディスク(バックアップファイル）をドライブにセットするように指示してきます。

Mount new disk on drive.
New drive#?

そこで,新しいフロッピィディスク(または,すべてのデータを消してもかまわないフロッピィディスク)をセットし，そのドライブに割り当てられているドライブ番号を入力します。ドライブ2の場合ですと，

2 RETURN

と入力します。このときマスターファイルとバックアップファイルを作るフロッピィディスクの種類が合わない場合は，

Can't backup.

と表示されます。このときはマスターファイルとバックアップファイルのフロッピィディスクの種類が異なっているか，ドライブ番号の入力がまちがっています。そこで再び(2)へもどります。

(5) ミニフロッピィディスクのバックアップを行う場合は，次のメッセージが表示されます。

Do you need physical formatting(y/n)?

すでに物理フォーマットが行われているフロッピィディスクをバックアップファイルを作るドライブにセットした場合は

n [RETURN]

を入力します。すると(6)へ進みます。

新しいミニフロッピィディスクで物理フォーマットが必要な場合は,

y [RETURN]

を入力します。すると,

Physical formatting a disk on driveX
Sure(y/n)?

と表示して,物理フォーマットを行うドライブのドライブ番号を確認してきます。もし,まちがっていれば,

n [RETURN]

と入力します。するともう一度(3)からやり直します。正しければ,

y [RETURN]

と入力します。すると

Physical formatting a disk on drive X,

と表示して,物理フォーマットを行います。

(6) マスターファイルの入っているドライブとバックアップファイルを作るドライブのアクセス表示用 LED が順に点灯すると同時に,画面にコピーしているトラック番号が表示されます。

Copying track XX
(**"XX"** はとびとびの値のこともあります。)

(7)無事に終了すれば

Complete.

と表示されます。

(8) マスターファイルがシステムディスクの場合は,バックアップファイルのフロッピィディスクもシステムディスクになります。

8.9.4 setinfN80(IDセクタの書き換え)

「8.7.6 ID」で説明したように,**ID** セクタを書き換えることによ

り，**N$_{80}$DISK-BASIC** をスタートさせると同時に，ファイル数をセットし，任意のプログラムを自動的に実行できます。このために，システムディスクには「**ID** セクタ書き込みプログラム（**setinfN80**）」が準備されています。以下，順を追ってこのプログラムの使い方を説明します。

(1) まず，**setinfN80** をロードします。

load "<ドライブ番号> setinfN80" **RETURN**

(2) プログラムを実行させます。

f·5 あるいは **run** **RETURN**

(3) **How many files (0−15)?**

とたずねてくるので，同時にオープンしたいファイル数を入力します。例えば，3つオープンさせたい場合ですと，

3 **RETURN**

と入力します。

このとき，**−1** を入力すると，**ID** のファイル数には 255(FFH)がセットされますから，ファイル数はセットされず，スタートするたびにファイル数を尋ねるようになります。

(4) 次に

Text for init?

と表示されますから，**N$_{80}$DISK-BASIC** のスタートと同時に実行したい BASIC のテキスト（コマンドやステートメント）を入力します。例えば，**"testN80"** というプログラムをスタートさせたい場合ですと，

run "testN80" **RETURN**

と入力します。

もし，ここで入力した BASIC のテキストが長すぎた場合は，

Text can have a maximum of 253 characters

と表示してブザーを鳴らし，もう一度，BASIC のテキストの入力を行うように指示してきます。

(5) 次に，今入力したファイル数と BASIC のテキストを表示します。

今の場合ですと，

# of files to allocate is 3
Text for init is :
run "test. N80"

と表示し，続けて

Mount a system disk on drive #1.
Type RETURN if ready.

と表示してきますから，自動スタートさせたいシステムディスクをドライブ番号1のドライブにセットし，[RETURN] を入力します。

(6) すると，

All done.

と表示され，**ID** セクタに BASIC テキストを書き込み，実行を終了します。

これで，このシステムディスクを使用して **N80DISK-BASIC** をスタートさせると，スタートと同時に自動的に，プログラム **"testN80"** を実行します。(もちろん，このシステムディスクには，プログラム **"testN80"** があるとします。)

(注　意) (3)で，**−1** を入力した場合は，(5)で

# of files is asked when starting up

と表示します。

8.9.5 xfilesN80 (フロッピィディスク間のファイル転送)

「8.9.3」で説明した **"backupN80"** を使用する場合は，バックアップファイルは同じ種類のフロッピィディスクにしか作成できませんでした。ここでは，異なったフロッピィディスク間のファイルのバックアップを行うプログラムについて説明します。

ファイル転送プログラム (**xfilesN80**) の使い方

このプログラムは，片面倍密度ミニフロッピィディスク，両面倍密度ミ

ニフロッピィディスクおよび8インチ標準フロッピィディスクの間でどの方向にでも，全ファイルをコピーします。

(1) システムディスクからファイル転送をプログラムをロードします。

load "<ドライブ番号> xfilesN80 RETURN

(2) プログラムを実行させます。

f・5 または **run RETURN**

(3) 次のメッセージが表示されます。

Transfer files
From drive #?

(4) マスターファイルが入っているドライブ番号を入力します。ドライブ番号が2の場合ですと，

2 RETURN

と入力します。

(5) すると次のメッセージが表示されます。

To drive #?

(6) バックアップファイルを作りたいフロッピィディスクをドライブにセットします。このとき，このフロッピィディスクの種類は，マスターファイルの入っているフロッピィディスクと異なっていてもかまいませんが，ミニフロッピィディスクを使用する場合は，すでにフォーマッティング（システムフォーマッティング）されていなければなりません。

フロッピィディスクがセットできたら，そのドライブ番号を入力します。ドライブ番号が3の場合ですと，

3 RETURN

と入力します。

(7) 次に

Sure（y/n）?

と確認してきます。もし，今まで指定したドライブ番号がまちがっていれば **n RETURN** と入力して，(3)からやり直します。 ドライブ

番号が正しければ,

y **RETURN**

と入力します。すると,

Copying file XXXX 2→3

と表示して,ドライブ番号2のドライブに入っているフロッピィディスク上の全ファイルをドライブ番号3のドライブに入っているフロッピィディスクへ転送します。転送が無事に終了すると,

Complete.

と表示されます。

このプログラムは,ファイルを転送するだけですから,同じ名前のファイルがないかぎり,バックアップファイルを作るフロッピィディスクに前から入っていたファイルを消すことはありません。

また, DISK code は転送できません。

注 意 もし,バックアップファイルを作るフロッピィディスクの未使用のクラスタ数より,マスターファイルの入っているクラスタ数の方が大きい場合は,途中で

Disk full

というエラーメッセージが表示されます。このプログラムを実行する前に,転送できるか確認するようにしてください。

8.9.6 sysgenN80(DISK codeのコピー)

「8.9.3」で説明した **"backupN80"** というプログラムを使うと,フロッピィディスクに入っている内容をすべてコピーします。しかし,何らかの理由で DISK code だけコピーしたい場合には,システムディスクに入っている **"sysgenN80"** というプログラムを使います。

(1) **N80 DISK-BASIC** を スタートさせ, DISK code のコピープログラムをロードします。

load "<ドライブ番号>sysgenN80" **RETURN**

(2) **プログラム**を実行させます。

f・5 または **run** RETURN

(3) 次のメッセージが 画面に 表示されます。

Copy system
Copy from drive #?

(4) コピーするもとの DISK code の入っているフロッピィディスクがセットされているドライブ番号を入力します。ドライブ番号が1の場合ですと，

1 RETURN

と入力します。

(5) すると次のメッセージが表示されます。

to drive #

(6) DISK code の書き込みを行いたいフロッピィディスクをドライブにセットし，そのドライブ番号を入力します。ただし，このときミニフロッピィディスクへコピーする場合は，あらかじめシステム・フォーマッティングされてなければなりません。

ドライブ番号が2のドライブにセットした場合は，

2

と入力します。

(7) 次に，

Hit RETURN if you are O.K.

と表示されますから，セットしたフロッピィディスクがまちがっていないか，また，指定したドライブ番号がまちがっていないか確認をして，正しければ，

RETURN

を入力します。もし，まちがっていれば， STOP キーを入力してプログラムを止め，もう一度，(2)からスタートします。

(8) **working**

と表示され， DISK code がコピーされます。コピーが終了すると，

Complete.

と表示されます。

このプログラムは，DISK code だけをコピーするプログラムですので，他のプログラムはコピーされません。また，DISK-BASIC(N-BASIC のディスクモード）の DISK code をコピーすることもできません。

第9章
ターミナルモード

これまで，**MKII**を単独のシステムだけで使う方法について説明してきましたが，この章では，**MKII**をターミナルとして，使う方法を説明します。**MKII**をターミナルとして使うことによって，遠く離れたコンピュータとのやりとり，別の**MK II**や他のパソコンとのネットワーク，電話線を通して各種のコンピュータサービスをうけたり，など幅広い応用ができます。

9.1 RS-232Cインタフェース

MKIIは，ターミナル用のインタフェースとして，**RS-232C規格のインタフェース**を持っています。**RS-232C規格**は**EIA**(Electronic Industries Association)で規定された，シリアルデータのインタフェース規格で，日本でもJIS C6361-71として制定されています。**RS-232C規格**はターミナル用のインタフェースの他に，いろいろなI/O機器のインタフェースとして採用されています。

(参　照) ターミナル以外に**RS-232C インタフェース**を使う方法は**13章 入出力インタフェース**で説明します。

9.2 ターミナルモードの基本動作

ターミナルモードにおける **MKII** の基本的な動作は，次の2つです。

1. RS-232C チャネルから入力した文字を，ディスプレイ画面に表示する。

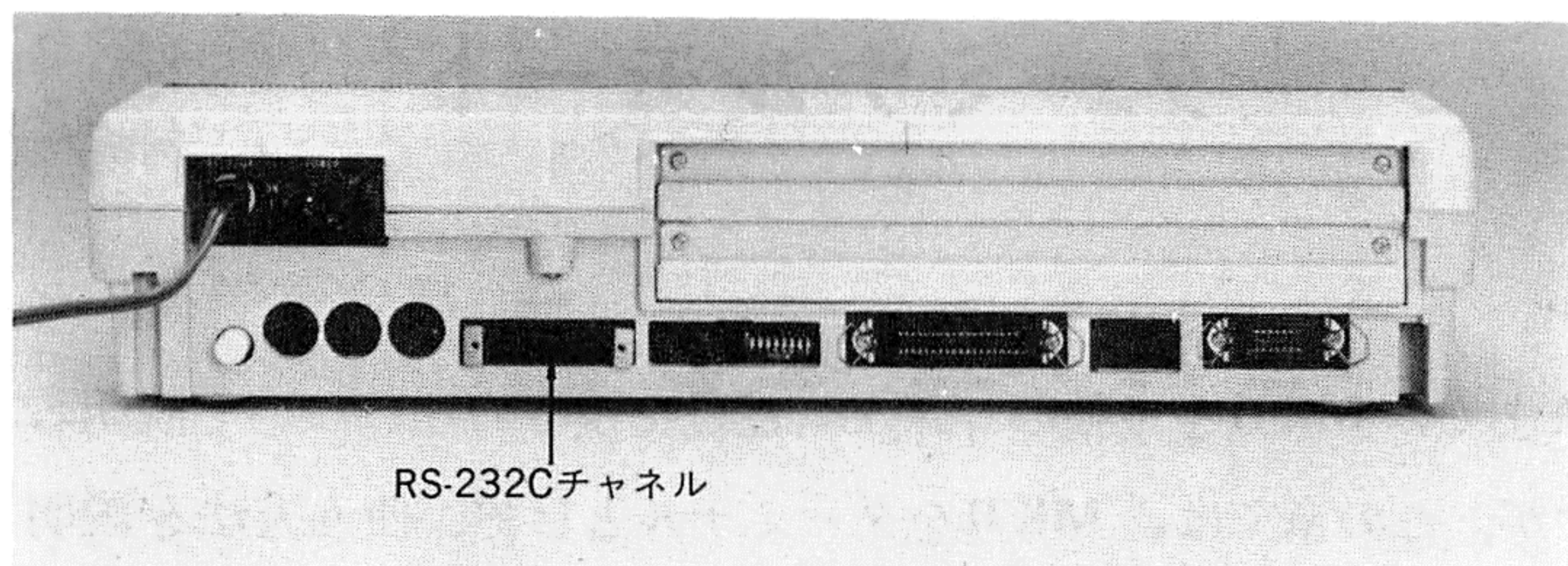
2．キーボードから入力した文字を RS-232C チャネルに出力する。

RS-232Cチャネル

図9．1　RS-232Cチャネル

PC-8001とくらべて **PC-8001** は RS232C コネクタを本体に備えていませんでした。したがってターミナルとして使うためには，RS-232C ケーブルユニット **PC-8062** が必要でした。**MK II** は本体に RS-232C コネクタがついています。

ターミナルモードに必要なハードウェアに関しては，図9．2のような関係が成り立ちます。

PC-8001　+　PC-8062　=　MK II

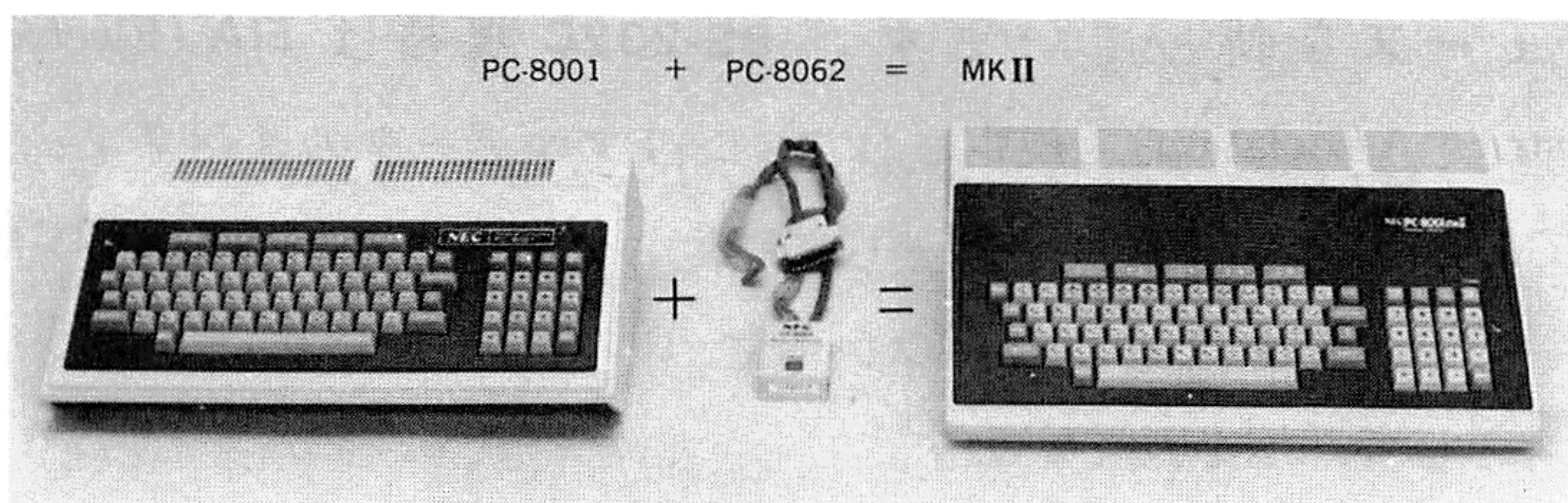



図9．2　ターミナルモードに必要なハードウェア

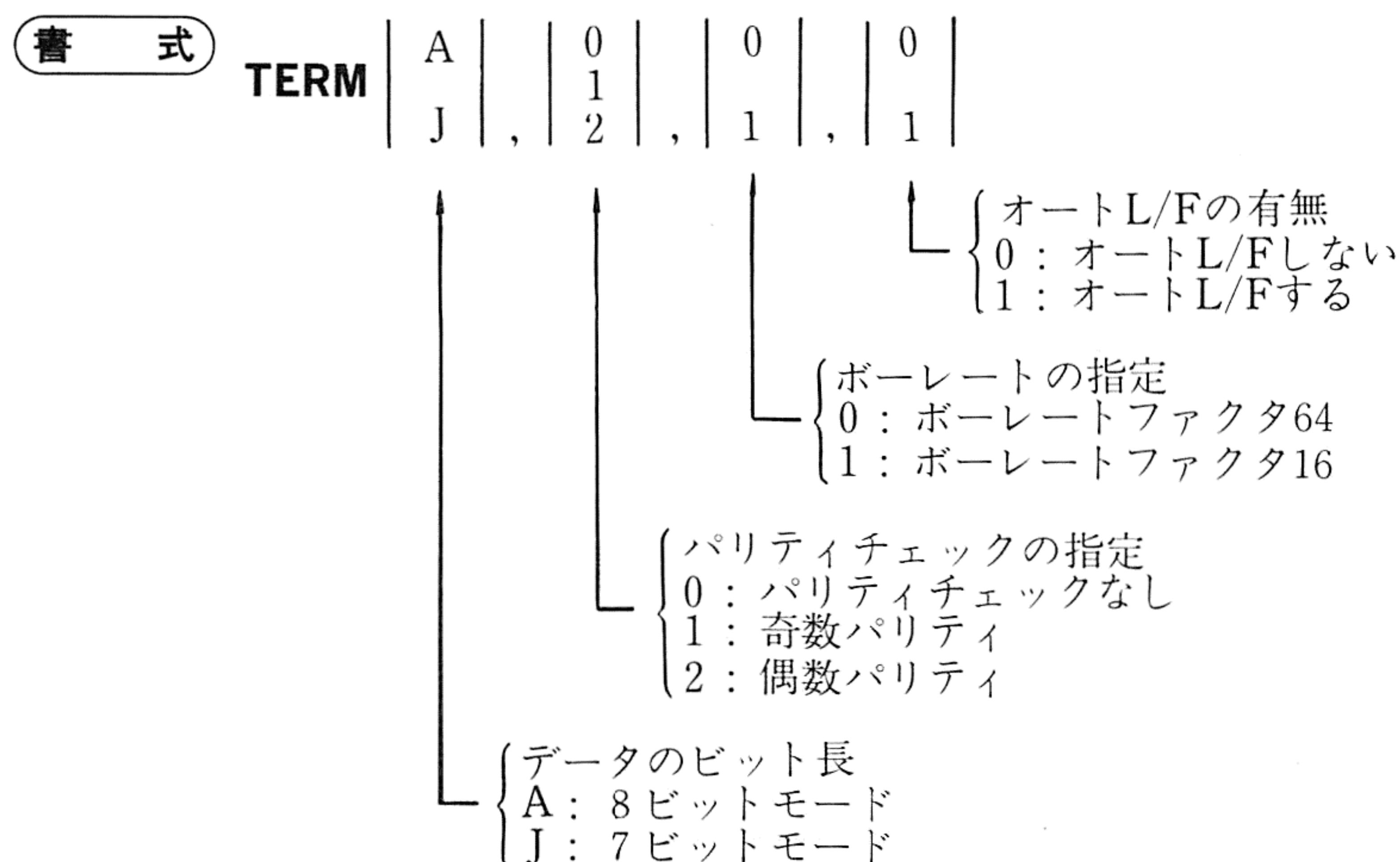
9.3　BASICモードとターミナルモードの切り換え方法

ターミナルモードへは，どの BASIC モードからでも同じように切り換えることができます。BASIC からターミナルモードに入るには

TERM コマンドを使います。ターミナルモードからBASICに戻るには，**GRPH** キーを押しながら **B** を入力します。ターミナルモードに入ってもBASICのプログラムはこわれません。

(PC-8001とくらべて) **PC-8001**では，ターミナルモードからBASICに戻るには **CTRL** キーを押しながら **B** を入力します

(書　式) **TERM** |A/J| , |0/1/2| , |0/1| , |0/1|

- オートL/Fの有無
  - 0：オートL/Fしない
  - 1：オートL/Fする
- ボーレートの指定
  - 0：ボーレートファクタ64
  - 1：ボーレートファクタ16
- パリティチェックの指定
  - 0：パリティチェックなし
  - 1：奇数パリティ
  - 2：偶数パリティ
- データのビット長
  - A：8ビットモード
  - J：7ビットモード

4つのパラメーターは，すべて指定しなくてはなりません。それぞれのパラメーターの意味を以下に示します。

(1) **データのビット長**

8ビットモードとは，最上位ビットで英数字とカナを切りかえるモードです。7ビットモードとは，SO(シフトアウト)コードがくるとそれ以後のデータをカナとみなし，SI(シフトイン)コードがくるとそれ以後のデータを英数字とみなすモードです。

(2) **パリティチェックの指定**

パリティチェックなしでは，パリティビットをつけません。パリティチェックを行う場合には，奇数パリティか偶数パリティのどちらかを指定します。

(3) **ボーレートの指定**

ボーレート（伝送速度）は，**MKII**の後部のジャンパスイッチと，このパラメーターとの組み合わせで指定します。

ボーレートファクタ 16 を指定した場合は，ジャンパスイッチでセ

ットされたボーレートの値が選ばれます。ボーレートファクタ64を指定した場合は，ジャンパスイッチでセットされているボーレートの1/4の速度になります。

出荷時にはジャンパスイッチは3にセットされていますから，ホストコンピュータの仕様に合わせて設定してください。

(注　意) ボーレートを600bps以上にセットした場合は，**MKII** の画面表示が間に合わなくなることがあるので注意してください。

(注　意) **MKII**のジャンパスイッチは，**PC-8801**と逆に並んでいます。

ジャンパスイッチ	ボーレート (bit per second) ボーレートファクタ16のとき	ボーレートファクタ64のとき
1	75	—
2	150	—
3	300	75
4	600	150
5	1200	300
6	2400	600
7	4800	1200
8	9600	2400

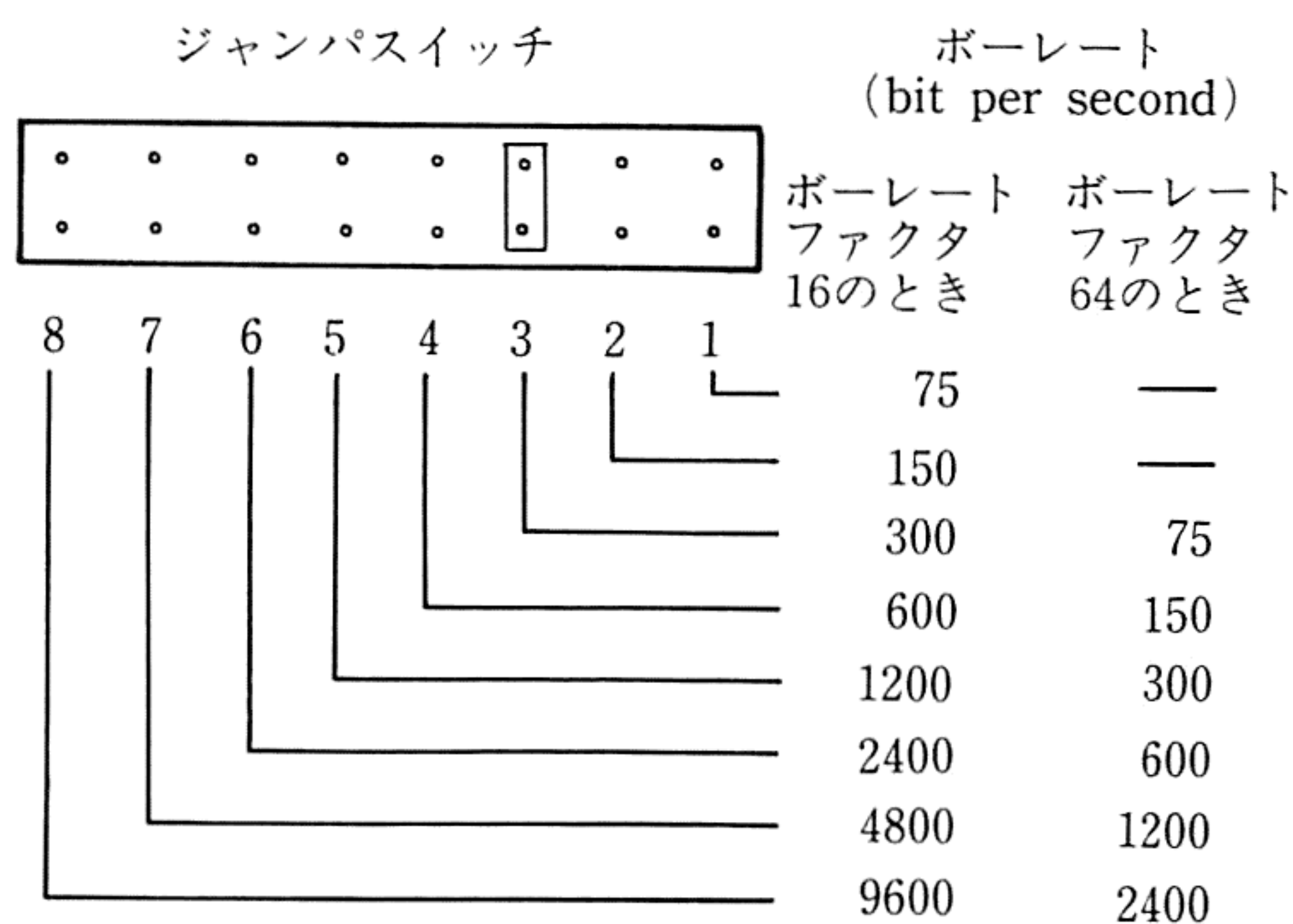



図9.3　ジャンパスイッチとボーレート

(4)　**オート L/F の有無**

オート L/F（ラインフィード）をするように指定すると，CRコードを受信したときに，改行も自動的に行います。

(Sample) 7 bitモード，偶数パリティ，ボーレートファクタ16，オートL/Fをするように設定する。

term j, 2, 1, 1

9.4　ターミナルモードでのファンクションキーの使い方

ターミナルモードに入ると，[f・6]～[f・10]（[SHIFT]＋[f・1]～[f・5]）は特

別の意味を持ちます（表9．1）。

（注　意）SHIFTキーを押してもファンクションキーの画面表示はf・1～f・5と変わらなくなります。f・6～f・10を押した場合，特別の機能を持つようになります。

表9．1　ターミナルモードでのファンクションキーの機能

キー	意　　味	ターミナルモードに入ったときの設定
f・6	コントロールコードの表示をするかどうかの切り換え。	コントロール コードを表示しない
f・7	半二重（オートエコー）と全二重（オートエコーなし）の切り換え。	全二重
f・8	プリンタにエコーするかどうかの切り換え。	プリンタに エコーしない
f・9	ディスプレイ画面をプリンタにコピーする。	——
f・10	プリンタのラインフィードを行なう。	——

9.5　ターミナルモードの使用上の注意

電電公社の TSS サービスやその他のオンラインサービスなどで，使用するときに注意していただきたいことをあげます。

1．7ビットモードでSOコードを受信してカナモードに入ったとき，**MK II** から，コントロールコードを送信することはできません。

2．カナキーをロックした状態で，**MK II** から，コントロールコードを送信することはできません。コントロールコードを送信するときはカナキーのロックを解除してから行ってください。

3．7ビットモードでは，SO コードを受信すると，SI コードが送られてくるまでは，ずっとカナモードになっています。

ホストによってはカナコードにすべて SO コードをつけ，SI コー

ドを送信しないものがありますから，この場合は，英数字モードに戻れなくなってしまいます。

第10章
モニタ

この章では，**MKII** で直接機械語を扱う場合に使用する，モニタ*の説明をします。BASIC だけを使っているユーザには，この章を読む必要はありません。機械語のプログラムの作り方は **N80-BASIC リファレンスマニュアル**をごらんください。

BASICモードからモニタに入るには**MON**コマンドを使います。

10.1 モニタのコマンド

モニタは表 10．1 に示す 8 種類のコマンドを持っています。

表10．1　モニタのコマンド一覧表

コマンド	意	味
D	Dump memory	メモリの内容をディスプレイに表示します。
S	Set memory	メモリの内容を変更します。
G	Go	機械語プログラムの実行をします。
W	Write tape	メモリの内容をカセットテープにセーブします。
LV	Verify tape	カセットテープの内容とメモリの内容を比較します。
L	Load tape	カセットテープの内容をメモリへロードします。
TM	Test memory	メモリのテストを行ないます。
CTRL +B	Back to BASIC	BASICモードに戻ります。

---

* この章で使う「モニタ」という用語は，直接機械語を扱うシステムプログラムの名称です。

10.2 モニタのコマンドの入力のしかた

モニタでは，コマンドの入力のしかたが BASIC とは異なります。BASIC と異なる点を列挙します。

1．大文字，小文字の区別がなく，小文字は大文字に変換されて表示されます。

2．スクリーン エディットができません。

3．INS DEL キーでコマンドを修正することはできません。コマンドの入力を誤った場合は，STOP キーを押して取り消します。

4．数値は，すべて 16 進数で表現します。 BASIC のように “&H” をつけて入力する必要はありません。

5．アドレス（0H～FFFFH）を入力した場合，下位の 4 行が有効になります。

6．メモリの内容（0H～FFH）を入力した場合，下位の 2 桁が有効になります。

7．入力可能なキーが限定されており，誤ってキーインすると，エラーになります。エラーメッセージは “？” です。

10.3 ユーザエリア

モニタで機械語のプログラムを作る場合は，ユーザエリアの中に書き込まなければなりません。それ以外のエリア(図10.1でアミをかけてある部分)は,BASICが使っていますから,ユーザが使ってはいけません。ここに機械語のプログラムを書き込むとBASICモードに戻ったとき,動作が保障されません。

ユーザエリアを BASIC のプログラムにだけ使う場合は，ユーザエリアを考える必要は全くありません。 BASIC でプログラムを作ると，図 10．1 に XXXX で示したアドレスが，上位に上がっていきます。

機械語でプログラムを作る場合は，何番地から，何番地までを使うかを

意識しなければなりません。図10．1の XXXX は，BASIC のプログラムと機械語の境界をあらわします。機械語を使う場合は，XXXX 番地から機械語を書き込むということを，**CLEAR** コマンドで，宣言しなければなりません。

CLEAR 300, &HXXXX−1

ユーザエリアの上限アドレスを図10．1に YYYY で表わしました。ROM-BASIC モードでは，YYYY は EA00H（**N-BASIC**）または E600H（**N80-BASIC**）に固定されています。DISK-BASIC モードでは，YYYY は可変です。YYYY の値は，EF54H 番地（下位）とEF55H 番地（上位）に書かれています。

DISK-BASIC

FFFFH ワークエリア

DISK code

YYYY

XXXX BASICのプログラムとデータ

8020H 割り込みベクトルエリア

8000H

ユーザエリア

ROM-BASICモード

ワークエリア

YYYY EA00H (N-BASIC) E600H (N80-BASIC)

XXXX BASICのプログラムとデータ

8020H 割り込みベクトルエリア

8000H

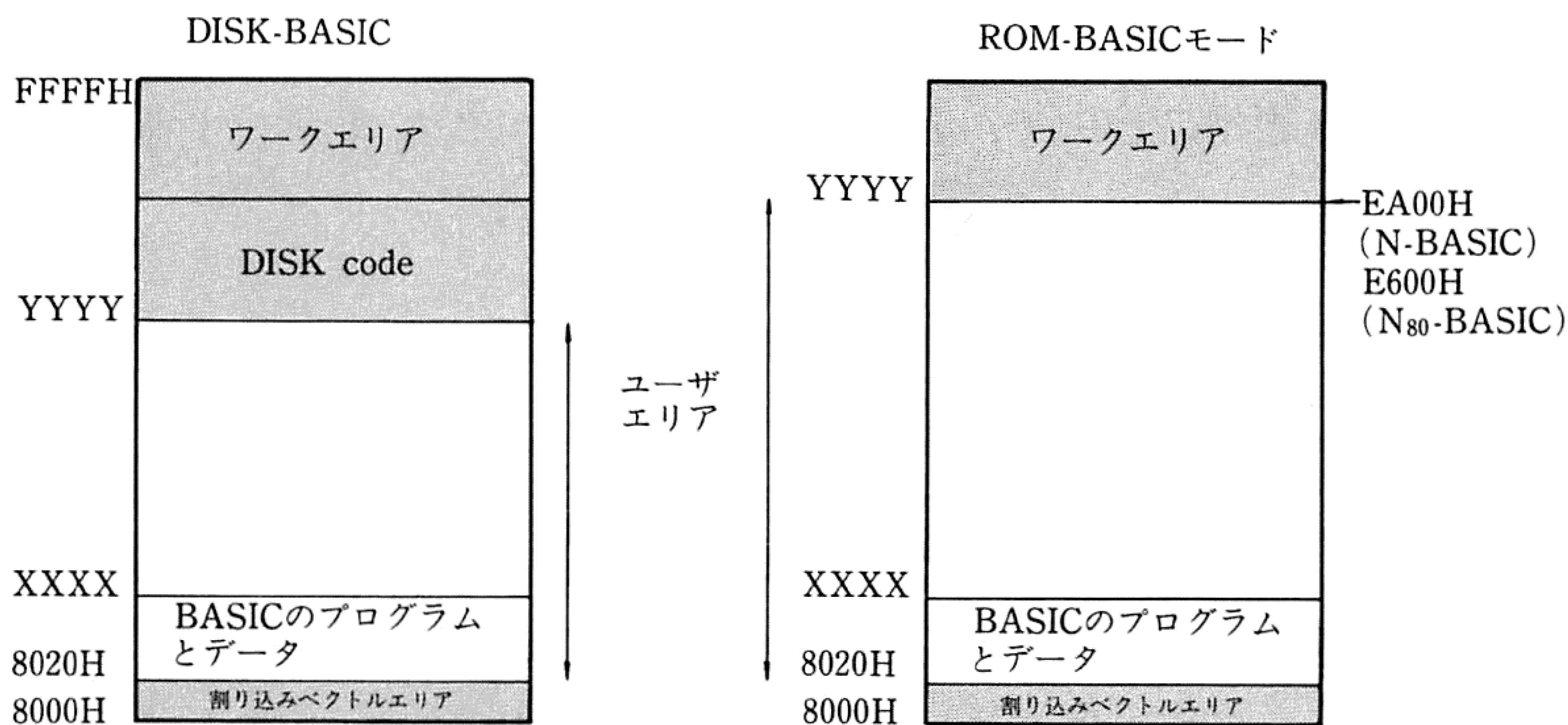



図10．1　ユーザエリア(アミのかかっていない部分)

10.4 各コマンドの説明

1．D (Dump Memory) メモリ内容の表示

(書　式) **D 〈スタートアドレス〉,〈エンドアドレス〉**

(Sample 1) **＊ DC000, C00F** [RETURN]

```
C000  00  01  02  03  04  05  06  07
C008  08  09  0A  0B  0C  0D  0E  0F
＊
```

(Sample 1) では，C000H 番地から，C00FH 番地までのメモリの

内容を表示させています。〈スタートアドレス〉，〈エンドアドレス〉は，5桁以上入力したときは，下位の4桁が有効になります。

Sample 2　*DC000 RETURN

```
C000  00  01  02  03  04  05  06  07
C008  08  09  0A  0B  0C  0D  0E  0F
*
```

Sample 2 では〈エンドアドレス〉が省略されていますが，結果は Sample 1 と同じになります。このように〈エンドアドレス〉を省略すると，スタートアドレスから16バイトだけ表示されます。

データ表示の途中で，スクロールを停止する場合は，ESC キーを入力します。スクロールの停止を解除する場合は，もう一度 ESC キーを入力します。

データ表示の途中で，表示を中止し，モニタのコマンド入力状態に戻したいときは，STOP キーを入力します。

2．S（Set Memory）メモリ内容の変更

(書　式) **S**〈変更のスタートアドレス〉

Sコマンドを入力すると，現在のメモリの内容を表示して，新しくセットするメモリ内容の入力待ち状態になります。この状態では，次のようなキー入力をすることができます。

(1)　変更データ

　2桁の16進数（00～FF）

(2)　CTRL ＋ H または INS DEL

　1つ前のアドレスにもどります。

(3)　スペース

　現在のメモリ内容を変更する必要のないときは，スペースを入力すると，次のアドレスに進みます。

(4)　RETURN または STOP

　Sコマンドを終了させます。

これ以外のキーを入力すると，エラーとなり，**S**コマンドからぬけ

出して，モニタのコマンド入力待ち状態に戻ってしまいます。

RETURN でSコマンドを終了したとき，または，エラーでSコマンドから，ぬけ出した場合は，S RETURN と入力するだけで，(〈変更のスタートアドレス〉を省略しても)中断したアドレスから，処理を続けることができます。

Sample 3 Sコマンドを使って，簡単なプログラムを作る例です。同じようにキー入力してみてください。次項で説明するGコマンドで実行することができます。

XX で示しているのは，現在のメモリ内容です。

```
*SE000 RETURN
E000   XX-21  XX-09  XX-E0  XX-CD
E004   XX-ED  XX-52  XX-C3  XX-66
E008   XX-5C  XX-43  XX-6F  XX-6E
E00C   XX-67  XX-72  XX-61  XX-74
E010   XX-75  XX-6C  XX-61  XX-74
E014   XX-69  XX-6F  XX-6E  XX-73
E018   XX-21  XX-00  XX- RETURN
```

Sample 3 でメモリにセットした値は，ひとつひとつが **MK II**に使われているマイクロプロセッサ **μPD780** の命令なのです。機械語のプログラムを使うには，**μPD780** の命令を理解することが必要です。

参　照　μCOM-82 ユーザーズ・マニュアル (**Bit-INN** で販売しております。)

3．G (Go) 機械語プログラムの実行

書　式　**G 〈開始アドレス〉**

Sample 4　***GE000 RETURN**

Sample 4 では，E000H 番地から，機械語プログラムを実行させています。E000H 番地以降に，でたらめが入っていると，**MK II**は，でたらめのとおりに動き，どうしようもなくなります。そんなときは，3．7 節に説明したようにウォームスタートを行ってください。ウォー

ムスタートできない場合は，コールドスタートしてください。

4．W（Write Tape）メモリの内容をカセットテープにセーブ

(書　式) W〈スタートアドレス〉，〈エンドアドレス〉

(Sample 5) *WE000, E019 [RETURN]

(Sample 5)では，E000H 番地から，E019H 番地までのメモリの内容をカセットテープにセーブしています。〈スタートアドレス〉,〈エンドアドレス〉は，5桁以上入力したときは，それぞれ下位の4桁が有効になります。[RETURN]キーを押す前に，テープレコーダに，カセットテープをセットし，[REC]ボタンと[PLAY]ボタンを押して用意しておかなくてはなりません。カセットテープにデータを書き込んでいる間は，カーソルが，その行の左端で点滅していますが，終了すると，*を表示して，モニタのコマンド入力待ちの状態に戻ります。

5．LV（Verify Tape）カセットテープの内容とメモリの内容の比較

(書　式) LV

LVコマンドは，**W**コマンドでセーブしたあとに，正しくセーブされているかどうかを確認するためのコマンドです。**LV**コマンドを入力する前に，データのセーブされている部分まで，カセットテープを巻き戻し，[PLAY]ボタンを押して用意しておかなければなりません。

カセットテープからデータを読んでいる間は，カーソルがその行の左端で点滅していますが，終了すると*を表示して，モニタのコマンド入力待ちの状態になります。

テープリードエラーもしくは，データ比較エラーのときは？を表示して，コマンド入力待ちの状態になります。

6．L（Load Tape）カセットテープの内容をメモリにロード

(書　式) L

Lコマンドは，**W**コマンドでカセットテープにセーブしたデータをロードするためのコマンドです。

カセットテープからデータを読んでいる間は，カーソルがその行の

左端で点滅していますが，終了すると＊を表示して，モニタのコマンド入力待ちの状態に戻ります。

テープリードエラーのときは，？を表示して，モニタのコマンド入力待ちの状態になります。

7．TM（Test Memory）RAMのテスト

（書　式）**TM**

TMコマンドは**MKII**の32Kバイトの**RAM**が正常であるかのテストを行うためのコマンドです。**TM**コマンドを実行すると**RAM**に書き込まれていたプログラムや，データはこわれてしまいますから注意してください。

TMコマンドの実行には，約3分30秒かかります。実行中に画面に文字等が出力されますが別に不良ではありません。

RAMが正常の場合は，起動時と同じ状態になります。

RAMに異常のあった場合は，ブザーが鳴りっぱなしになって，処理が停止します。このときには，リセットスイッチを押して，もう一度**MON**コマンドでモニタを動かして**RAM**のどこに異常があったかを調べてください。不良のアドレスとその内容は，次のように，記録されています。

・不良のメモリアドレス	FF39H 番地（アドレスの下位）
の格納アドレス	FF3AH 番地（アドレスの上位）
・不良のメモリに書いたデータ	FF3BH 番地
・そのメモリから読み出されたデータ	FF3CH 番地

（注　意）メモリのテストは，FFFFH番地から行いますから，**TM**コマンドを入力してから数秒でエラーになった場合は，FF39H～FF3CH番地の内容が正常でない場合が考えられます。

そのような場合は，ユーザがこの部分への読み書きを行って調べる必要があります。

8. ^B (Back to BASIC) BASIC に戻る。

(書　式) **^B** (**CTRL** キーを押しながら **B** キーを押します。)

^B コマンドは，モニタから BASIC に戻るときに使います。

10.5 機械語のプログラムを使うときの注意

BASICだけを使ってプログラミングする場合（BASIC の**POKE, OUT, USER**など機械語を扱う命令を使わないとき）は，ユーザは，ユーザーズエリア，ワークエリアなどのメモリの管理をする必要は全くありません。ユーザは，Out of Memoryエラーが出たときに，メモリが足りなくなったことを知るぐらいで，通常は，メモリがどのように使われているかを気にする必要はないのです。一方，機械語のプログラミングをする場合，または，市販のソフトウェアで機械語が含まれているものを使用する場合は，注意が必要です。機械語のプログラムを使う方法として，次のようなものがあげられます。

① モニタの**S**コマンドを使ってプログラムを作ったり，**L**コマンドでプログラムをロードして，**G**コマンドで実行する。

② BASICの中で**CLEAR**によって機械語領域を宣言し，**POKE**や，**CMD BLOAD**などでメモリに書き込み，**USR**関数で実行する。

機械語のプログラミングの経験のある方はご存知のように，不完全な機械語のプログラムを実行させた場合，**MK II**の制御がきかなくなってしまう（キー入力ができなくなる）ことがよくあります。機械語のプログラムは，BASIC の管理下にありませんから，エラーがあったとしても，エラーメッセージを出力してコマンド入力待ちの状態に戻ってくることはないからです。（ユーザからみればプログラムの誤りでも，μPD780は，誤りと思わないで，正直に実行を続けます）。**MK II**の制御がきかなくなってしまうと，機械語のプログラムがどのように実行されているかは見当もつきません。突然コールドスタートがかかってしまう

こともあれば，メモリの内容をめちゃくちゃに書き換えて，BASICのワークエリアや，BASICのプログラムが書かれている部分，あるいは，機械語プログラム自身をこわしてしまうことも考えられます。

このような場合は，ウォームスタート(できなければコールドスタート)をかけます。ウォームスタートができても，上記のように，メモリが書きかえられた可能性がありますから，メモリがもとの状態に戻る，という保障はありません。

したがって，機械語のプログラムを作成する場合は，メモリ上のBASICのプログラムなどが，失われてもいいように，必ず，セーブしておいてから実行させるようにしてください。また，機械語のプログラムを使ったあと，他の作業(別のBASICのプログラムを使うときなど)を始める前に，リセットボタンを押して，メモリのイニシャライズをすることをお勧めします。

第11章
ハードウェア

11.1 外　　観

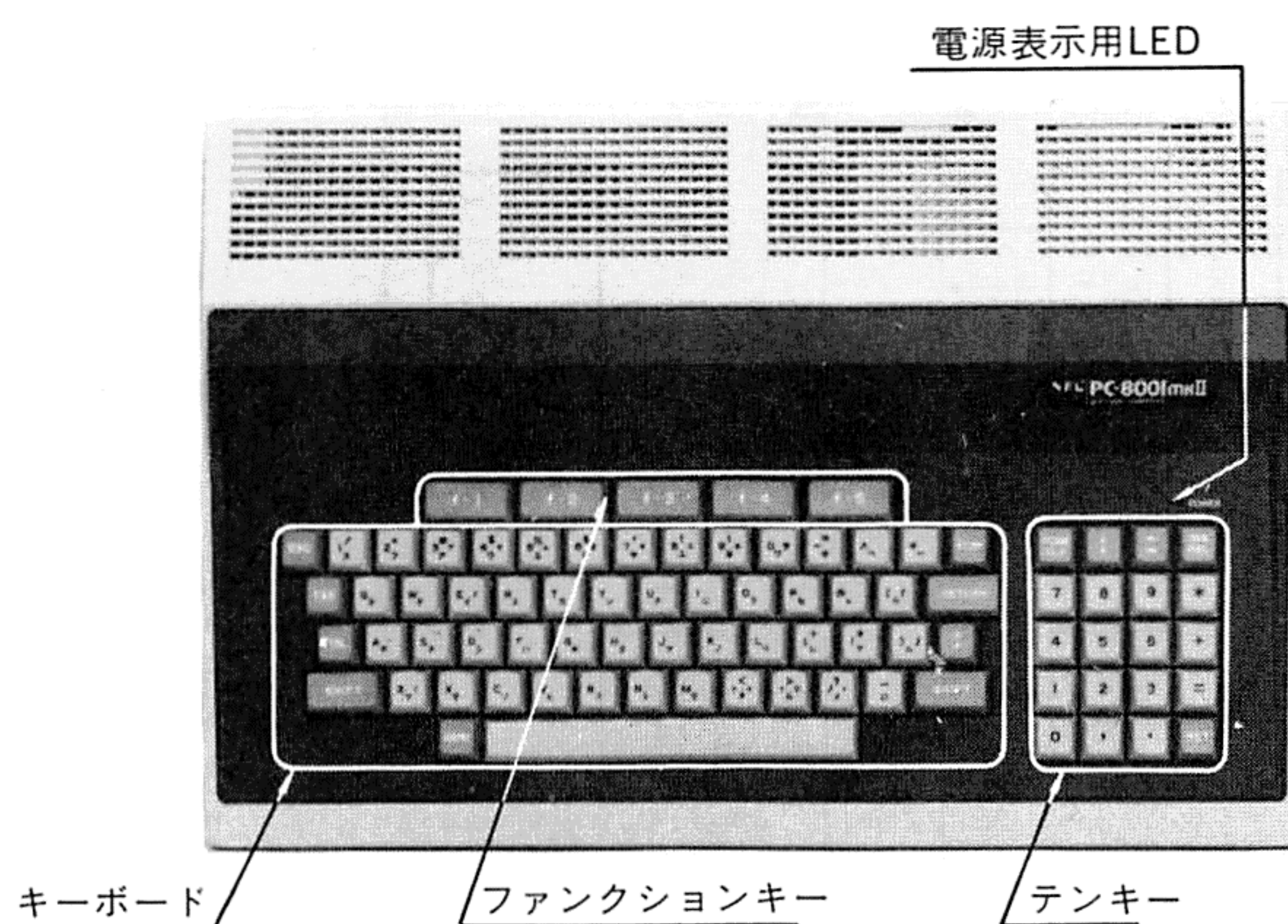



図11．1　本体表面

図11．2　本体背面

11.2 ブロックダイヤグラム

CPU
ROM 32K
N_{80}-BASIC
N-BASIC
RAM 64K
プリンタ インタフェース
CMT インタフェース
RS-232C インタフェース
PC-8031 インタフェース
汎用 I/O
漢字用 I/O
スロット 1
スロット 2
バス・スロット
PC-8881
PC-8882
DMAC
CRTC
キャラクタ コントロール
グラフィックコントロール ビデオインタフェース
グラフィック用 VRAM16K
割込み コントローラ
カレンダ 時計
キーボード
スピーカ
電源
+5V, +12V −12V
プリンタ
テープレコーダ
モデムまたは音響カプラ
ミニディスクユニット
拡張用ミニディスクユニット
バーコード リーダー
漢字ROM ボード
カラーモニタ
モノクロモニタ
ライトペン
RFモジュレータ
家庭用テレビ

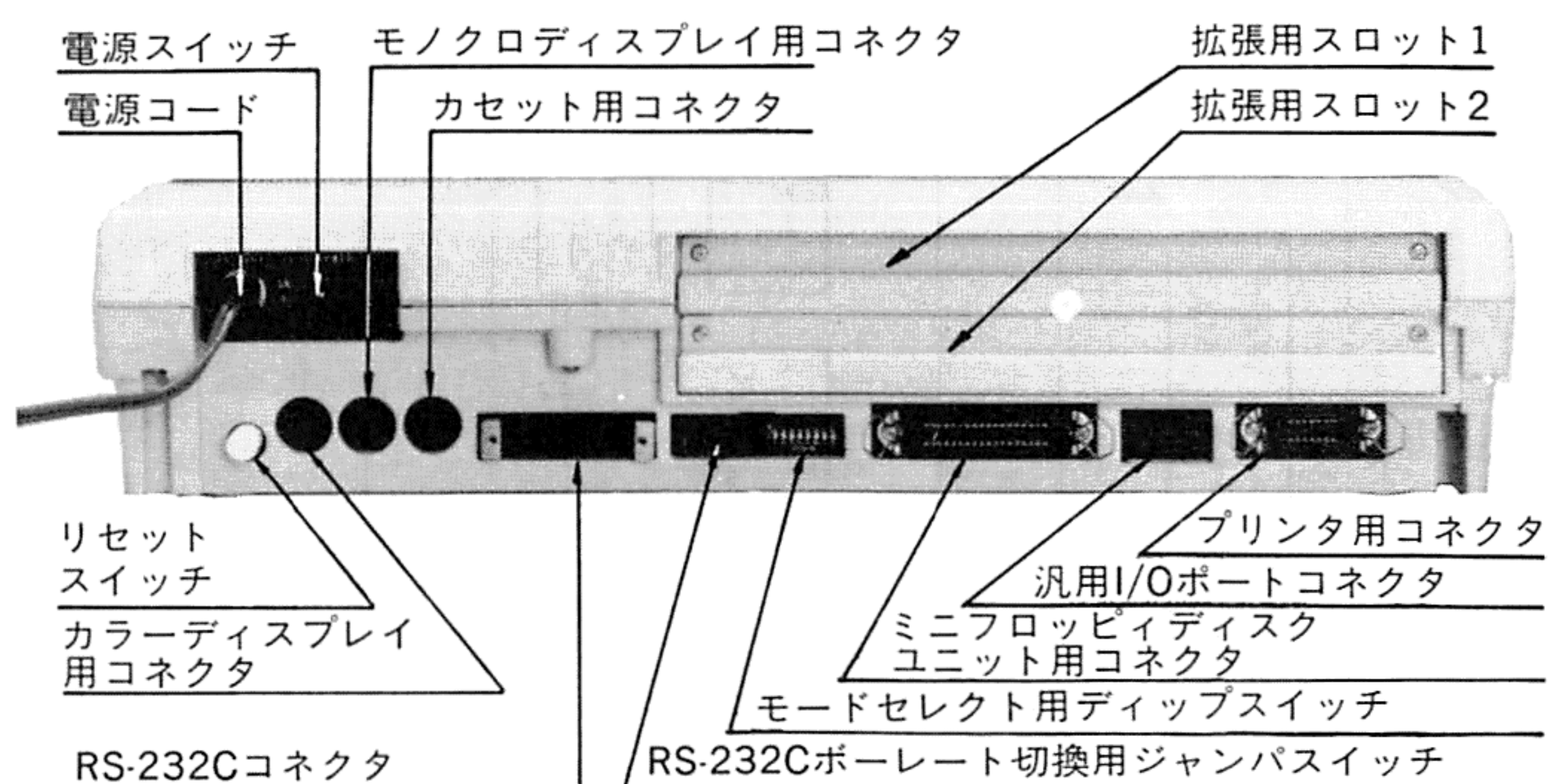
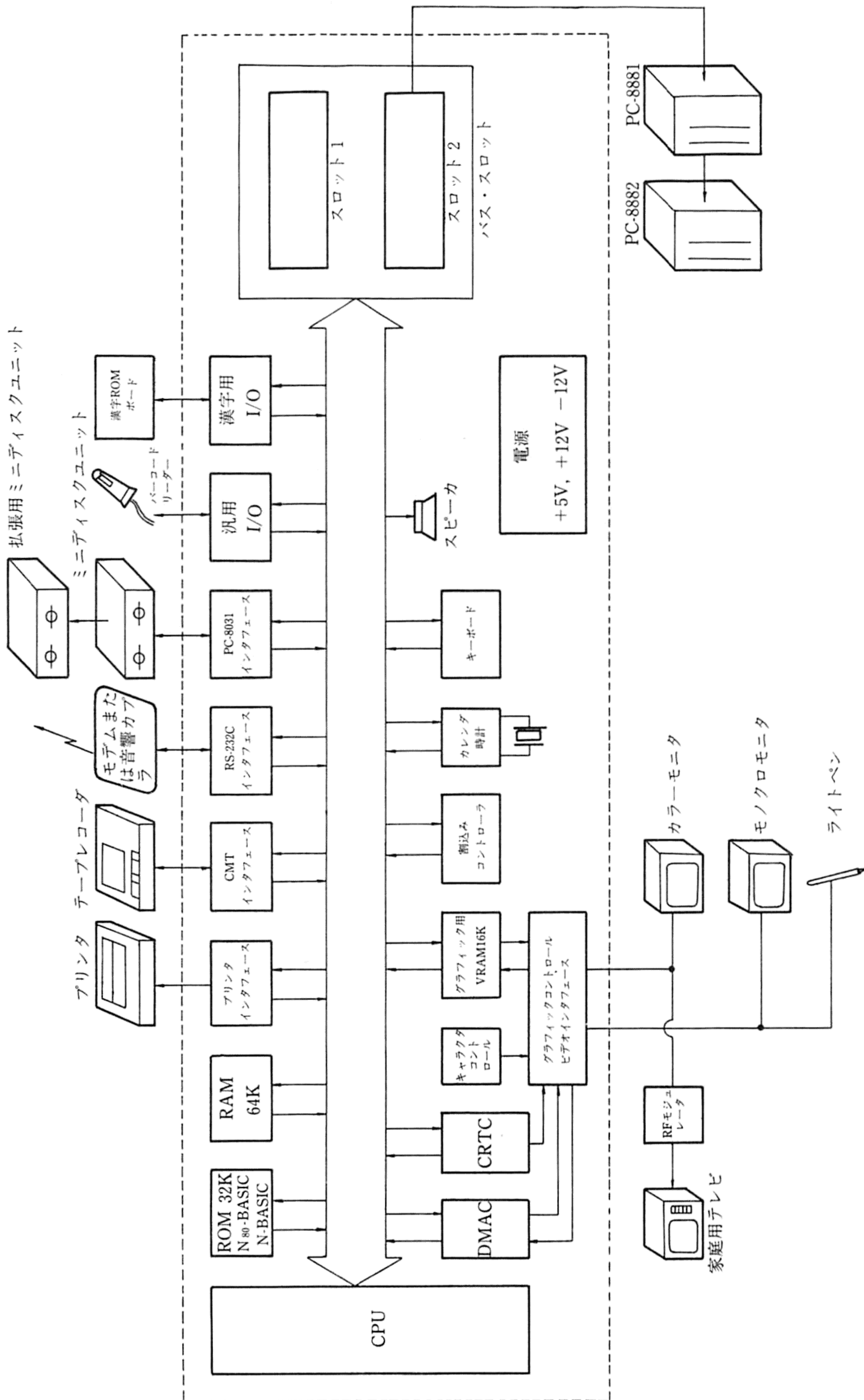



図11.3 ブロックダイヤグラム

11.3 機能仕様

(1) μ**PD780C-1**(Z-80Aコンパチブル), 4MHzクロック

(2) **ROM**

N_{80}-BASIC	32Kバイト(このうち 24Kバイトは N-BASICと共通)
本体内部増設可能	8K バイト

(3) **RAM**

BASIC 動作時	32K バイト
グラフィック用 VRAM	16K バイト
拡張 RAM 内蔵 (PC-8012-02 バンク 0 とコンパチブル)	32K バイト
スロット内増設可能	32Kバイト単位でバンク切換

(4) **表示能力**

テキスト表示

80文字×25行, 80文字×20行

72文字×25行, 72文字×20行

40文字×25行, 40文字×20行

36文字×25行, 36文字×20行

※上記いずれかを選択

リバース, ブリンク, シークレット (キャラクタ単位に指定可)

カラー8色 (キャラクタ単位に指定可)

グラフィック表示

モノクロモード	640×200 ピクセル
アトリビュートカラーモード	640×200 ピクセル (キャラクタ単位に8色の色指定ができる)
4色カラーモード 0	320×200 ピクセル4色カラーモード 黒, 赤, 緑, 選択色 (ピクセル単位に4色の色指定ができる)

4色カラーモード1	320×200 ピクセル4色カラーモード 青，マゼンタ，シアン，選択色 (ピクセル単位に4色の色指定ができる)

ビデオ出力

R.G.B セパレート出力方式（カラー）

コンポジットビデオ信号出力方式（輝度変調，モノクロ）

家庭用 T.V.（T.V. アダプタ経由）に接続可

(5) **漢字ROMボード（オプション）**

文字構成	16×16 ドット
文字種類	JIS 第一水準の漢字（2965 字） 非漢字（約 700 種）
画面構成	40文字×12行

(6) **キーボード**

JIS 標準配列準拠

テンキー，コントロールキー，5ファンクションキー

(7) **拡張用スロット**

2スロット(PC-8012，PC-8801 上位コンパチブル)

(8) **オーディオカセットインターフェース**

600bps

(9) **汎用 I/O**

入力2ビット，出力3ビット，割込 1ch

(10) **プリンタインタフェース**

パラレルインターフェース（セントロニクス社仕様に準拠）

(11) **シリアルインタフェース**

RS-232C 規格に準拠。割り込み／ポーリング制御可

75/150/300/600/1200/2400/4800/9600

(12) **8インチフロッピィディスクインタフェース**

本体内スロットに内蔵可

(13) **ミニフロッピィディスクインタフェース**

内蔵（PC-8031，PC-8031-2W 用）

(14) **ライトペン**

PC-8045 使用可

(15) **カレンダ時計**

NiCd 電池でバックアップ

(16) **電源**

AC100V±10%，50/60Hz

(17) **使用条件**

0～35°C，20～80%（但し結露しないこと）

(18) **外形寸法**

440(W)×295(D)×96(H)mm

(19) **消費電力**

20W

(20) **重量**

4kg

第12章
メモリマップ

MKIIは，次のようなメモリ構成になっています。

FFFFH RAM 64Kバイト 0H
7FFFH ROM2 16Kバイト 4000H
3FFFH ROM 1 16Kバイト 0H
7FFFH 6000H ROM 3 8Kバイト
BFFFH 8000H G-V RAM 16Kバイト
オプション RAM 32Kバイト

ROM 1, ROM2……N_{80}-BASIC
ROM 3……増設用ROM

図12.1　メモリマップ

FFFFH RAM 8000H 0H
ROM
G-VRAM

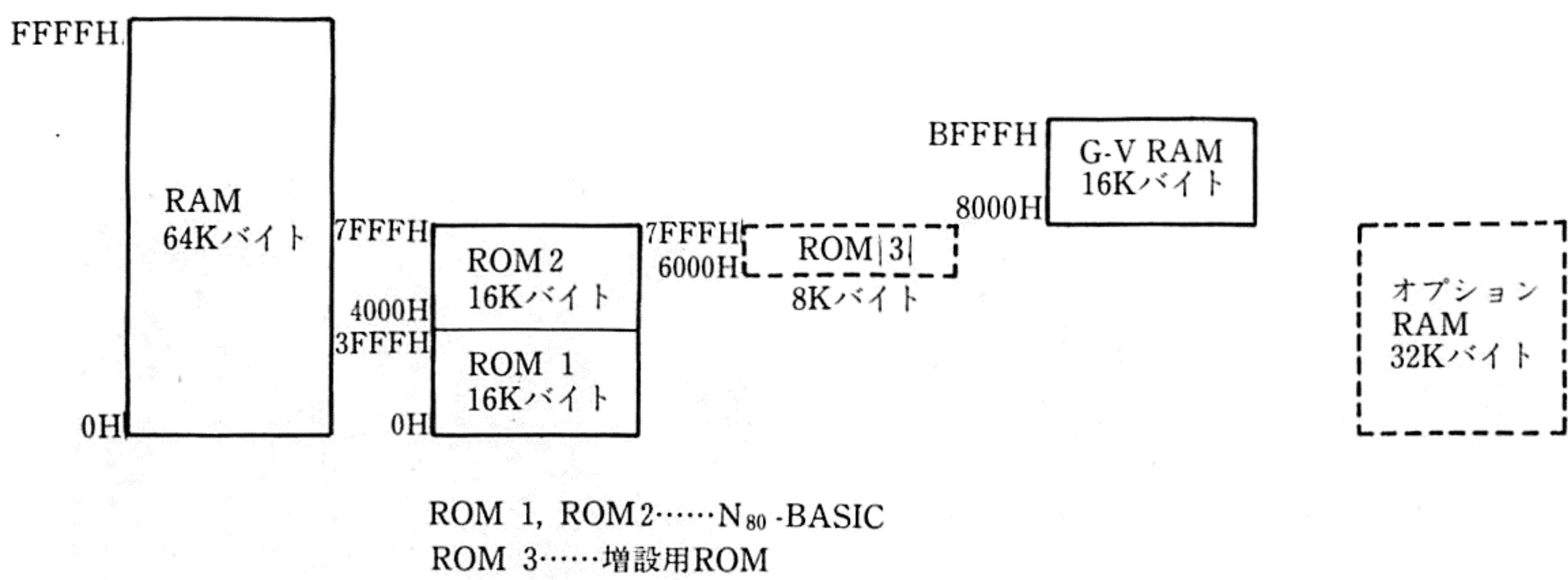
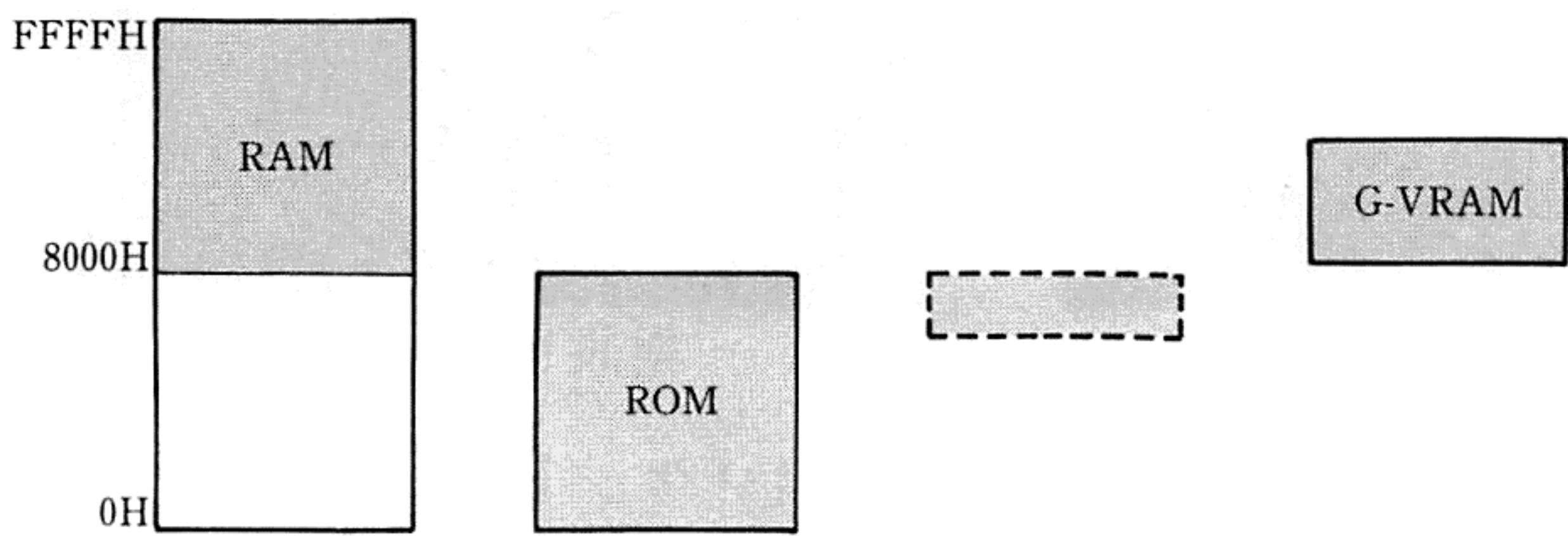



図12.2　N_{80}-BASICで使われるメモリ

FFFFH RAM 8000H 0H
ROM

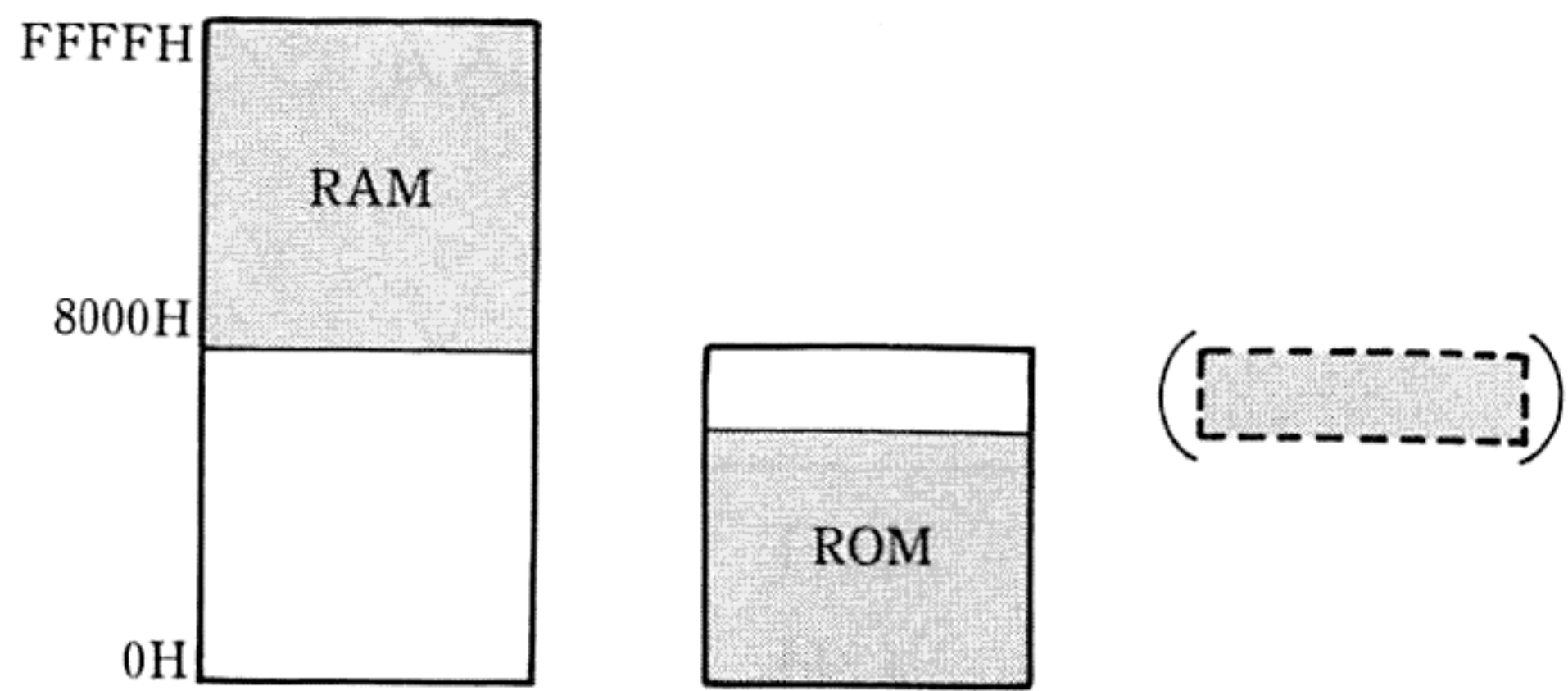



図12.3　N-BASICで使われるメモリ

12.1　増設ROMの使い方

MKII は PROM 増設用の IC ソケット（図 12.1 の ROM3）に PROM（8K バイトを増設することができます。次に示すメモリ素子を使用してください。

PROM　インテル社2764タイプ（アクセスタイム 300ns 以下）

（注　意）実装する場合は必ず電源を切ってからおこなって下さい。
電源を入れる前に PROM の向きを確認して下さい。

PROM増設用ソケット　　　　増設したところ

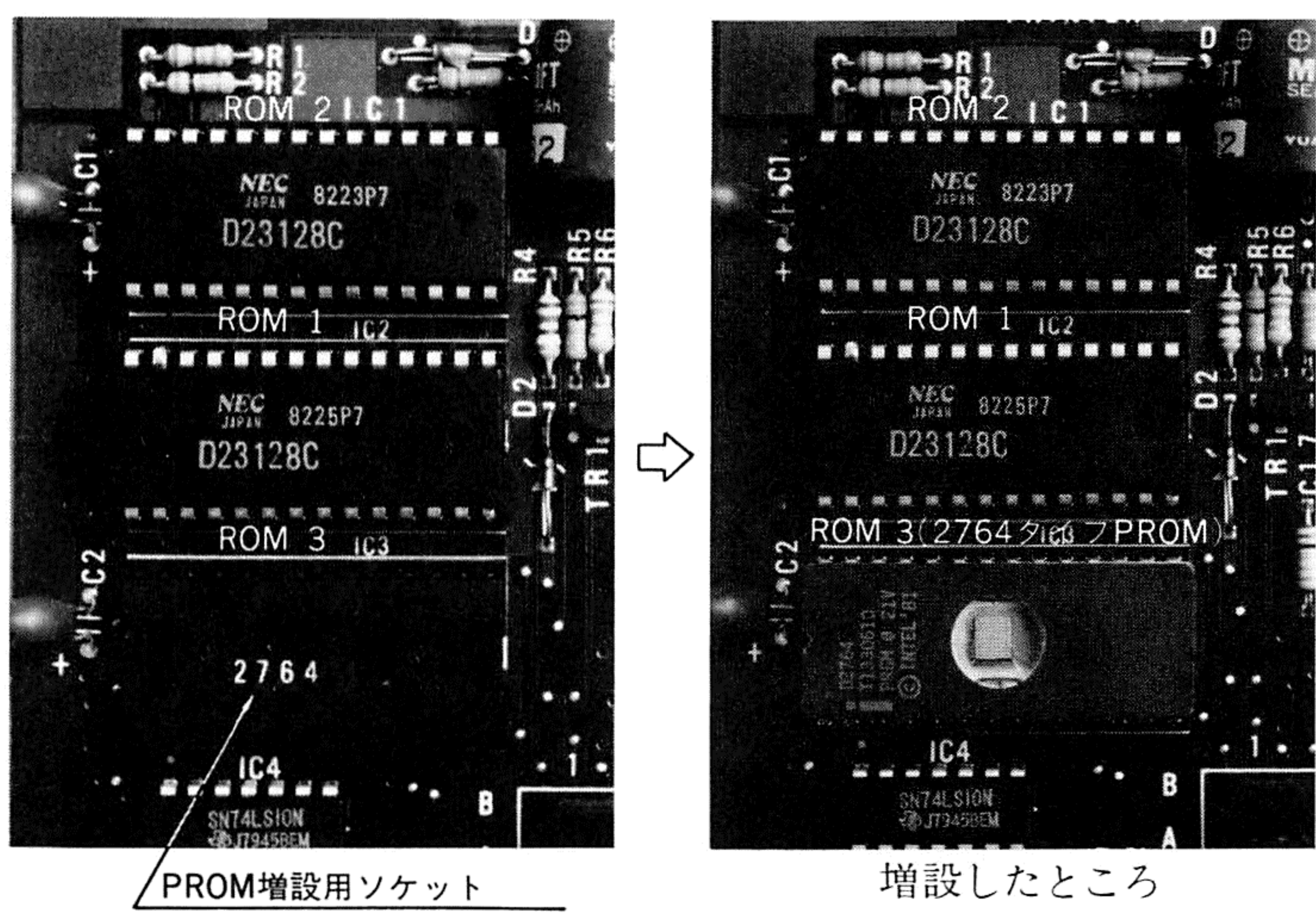



図12．4　PROM増設用ソケット

増設された PROM のアドレスは 6000H～7FFFH です。

増設ROMに書き込んだ機械語プログラムを**MKII** のリセット時に実行する必要のあるときは，増設ROMに次に示すデータを書き込んでください。

アドレス	N_{80} -BASIC	N-BASIC モード
6000H	41H	41H
6001H	42H	42H
6002H	00H	—

このようにセットされていると，**MKⅡ**をリセットしたときに**CALL 6002H**が実行されます。したがってリセット時に実行する必要のあるプログラムを，**6003H**(N_{80}-BASICモード）または**6002H**(N-BASICモード）番地から書いておけばよいのです。

リセット時に実行する必要がなければ6000H，6001H番地の内容は上記以外のデータにして下さい。

12.2 WAIT機能

MKⅡはCPUのM1サイクルに自動的に1WAITがかかります。そのため本体内または拡張用スロットに増設したROMにはアクセスタイム300ns以内のものを使用することができますが，これよりもアクセスタイムの遅いROMを使用する場合はさらにCPUのメモリ・リードサイクルに1WAITを追加することができます。次に示すいずれかの方法によります。

① ディップスイッチの2をONにする。

② 拡張用スロットの$\overline{\text{MWAIT}}$信号をLOWレベルにする。

12.3 RAMの増設方法

MKⅡは拡張用スロットに**PC-8012-02**（32Kバイト増設RAMボード），**PC-8801-02**(128Kバイト増設RAMボード）を実装することによりRAMを増設することができます。

(1) **PC-8012-02の増設方法**

MKⅡは，**PC-8012-02**のバンク0の機能を持っていますので**PC-8012-02**を増設する場合はバンク1，バンク2，バンク3のいずれかにして実装して下さい。

PC-8012-02

読出し制御

SW1	0	1	2	3
	使用禁止	バンク1	バンク2	バンク3

書き込み制御

SW2	0	1	2	3
	使用禁止	バンク1	バンク2	バンク3

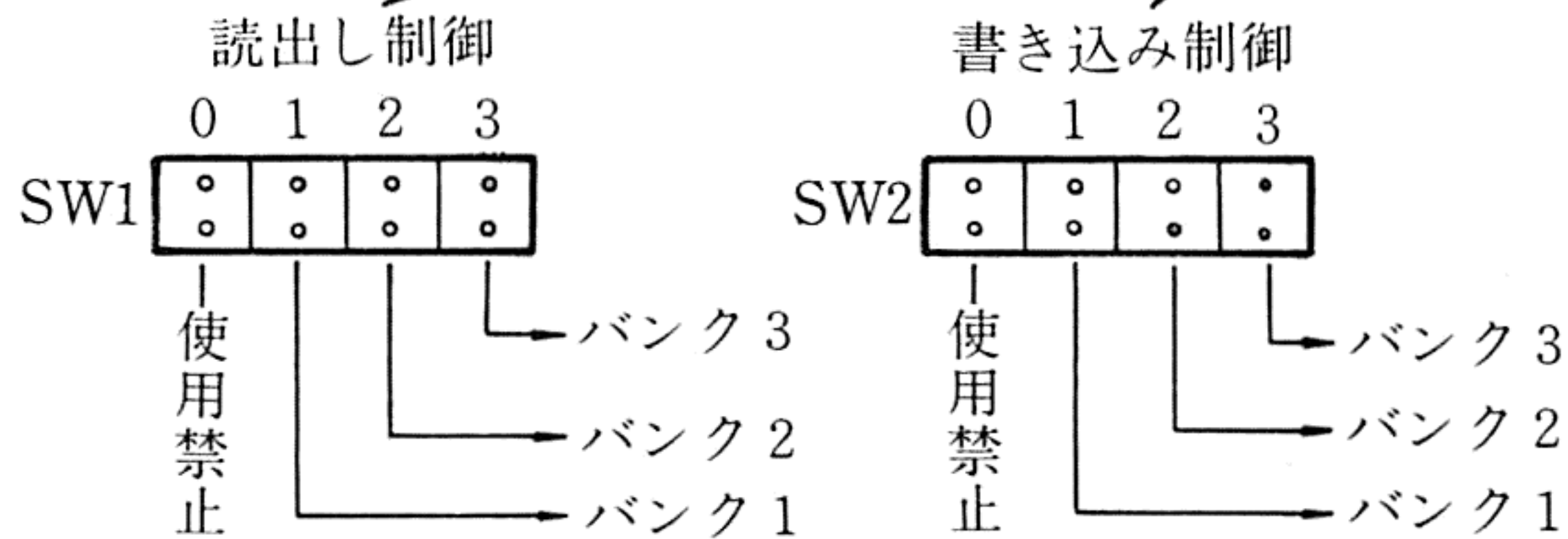
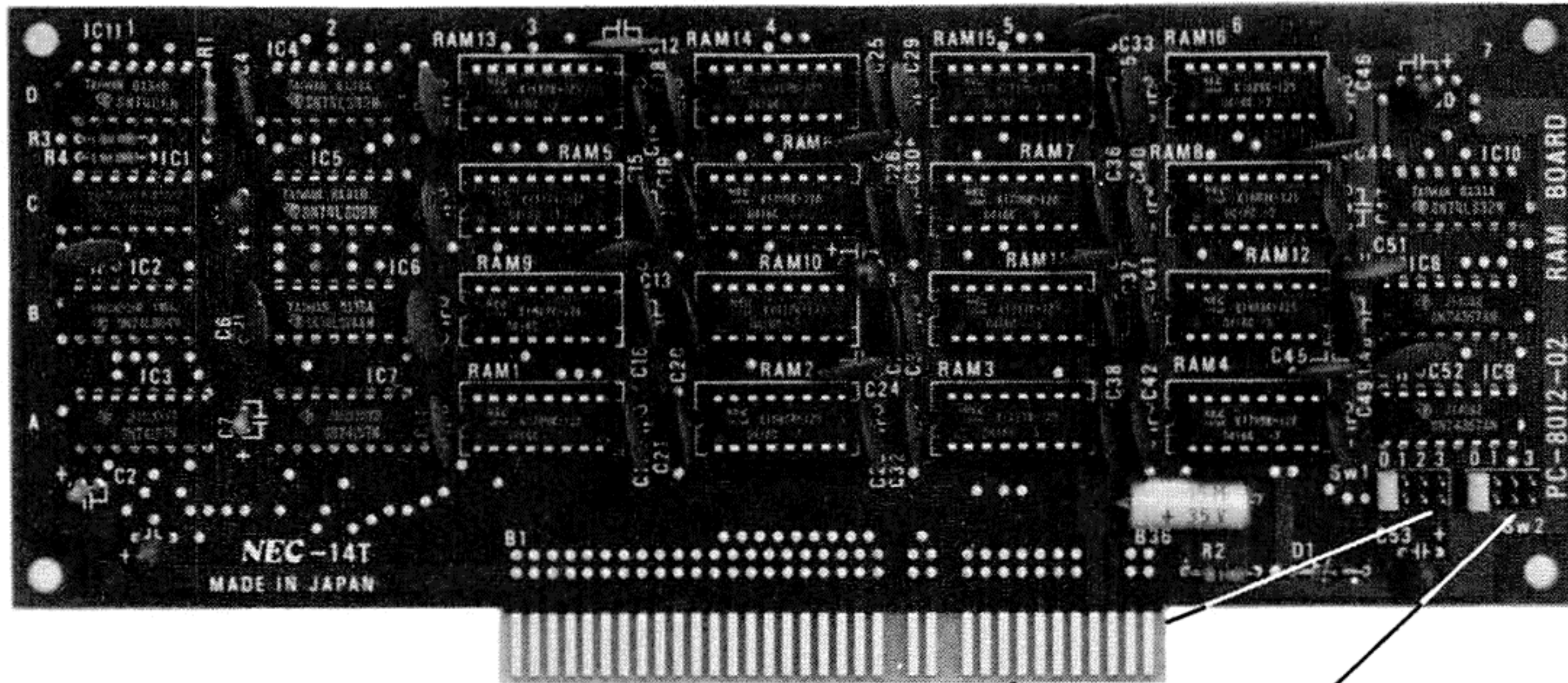



図12．5　**PC-8012-02**のジャンパスイッチ

PC-8012-02 をバンク 1 にして 1 枚実装した場合のメモリマップを図 12.6 に示します。

	①	②	③
FFFFH〜8000H	RAM 32Kバイト	RAM 32Kバイト	RAM 32Kバイト
8000H〜0H	ROM 32Kバイト N_{80}-BASIC	RAM 32Kバイト (バンク0)	増設RAM 32Kバイト (バンク1)
I/OアドレスE2H OUT DATA	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 1	0 0 1 0 0 0 1 0
		バンク0(本体内) 読み出し，書き込み許可	バンク1(増設) 読み出し，書き込み許可

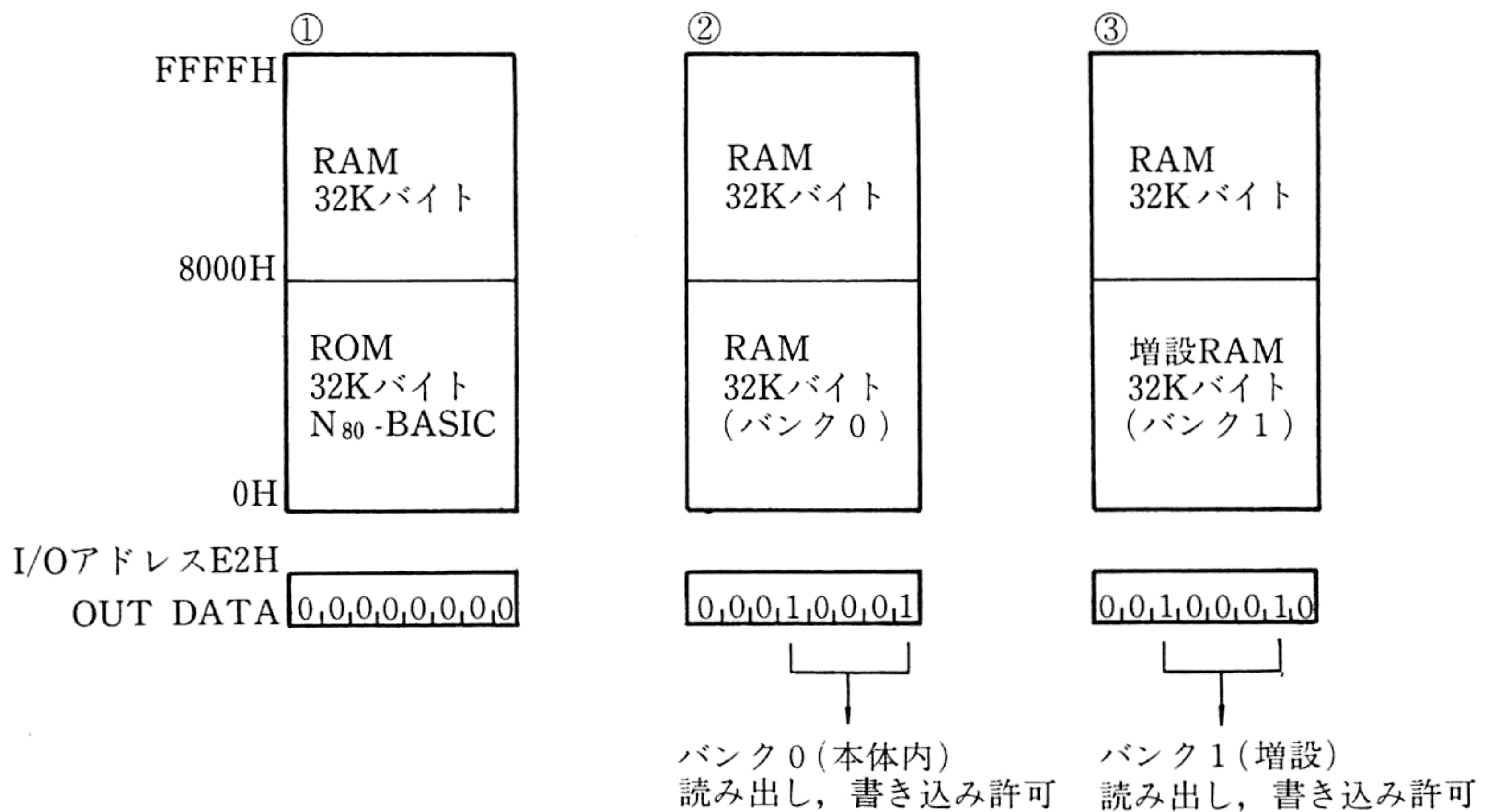



図12．6　増設RAMのメモリマップ1

(2) **PC-8801-02の増設方法**

MKII は **PC-8012-02** のバンク0の機能を持っていますので **PC-8801-02** を増設する場合は **PC-8801-02** のジャンパスイッチ JSW2 を1に切り換えて I/0 アドレス E2H ポートの読み出しを禁止して下さい。

JSW1
4 →カード3(バンク12～15)
3 →カード2(バンク8～11)
2 →カード1(バンク4～7)
1 →カード0(バンク0～7)

JSW2
1 →E2H読み出し禁止(MKIIに実装するとき)
2 →E2H読み出し許可(PC-8801, PC-8001に実装するとき)

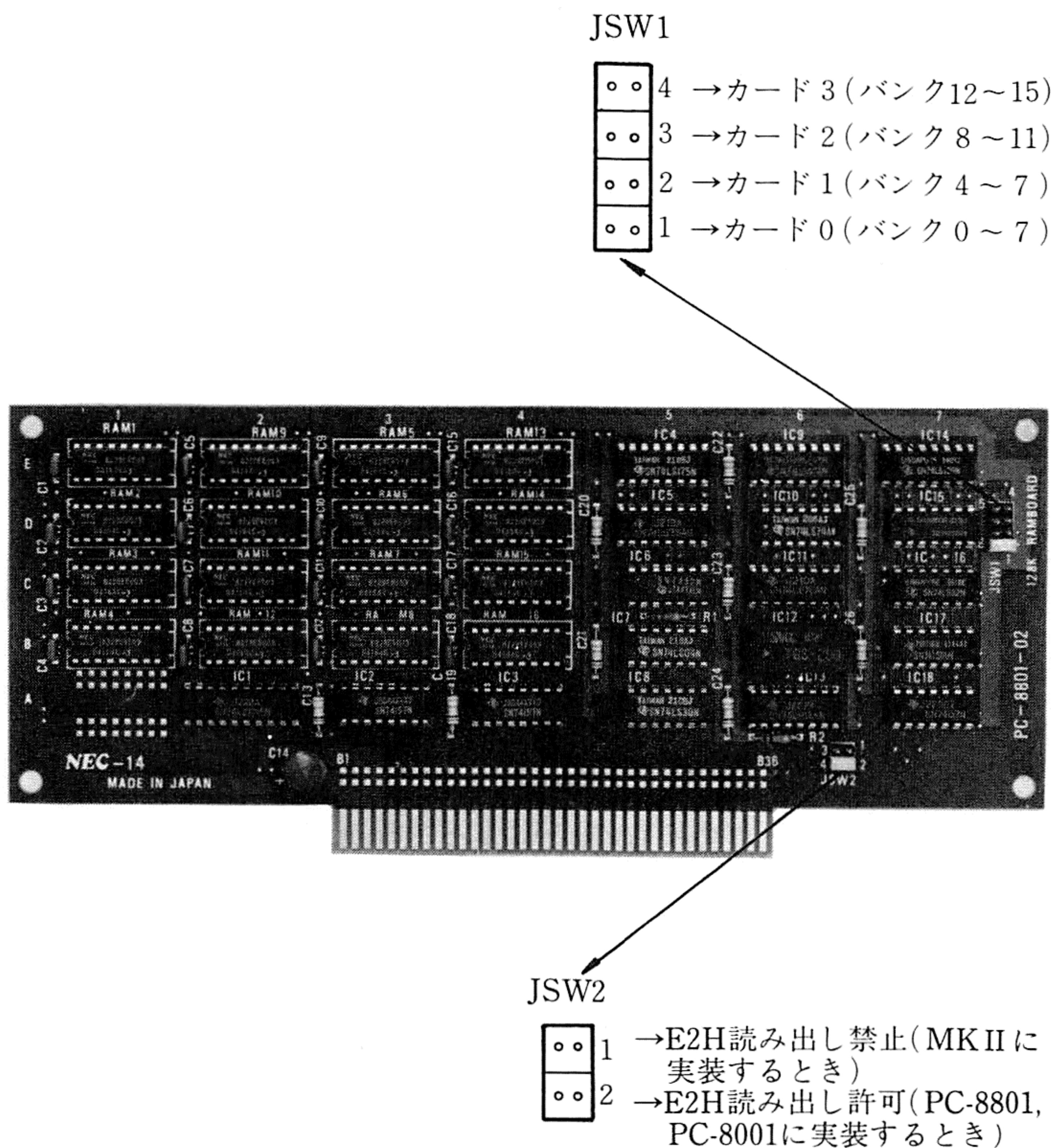



図12.7 PC-8801-02のジャンパースイッチ

PC-8801-02 をカード0にして1枚実装した場合のメモリマップを図 12.8 に示します。

(注 意) 実装されていないカードを指定した場合，本体内のバンクが選択されます。

	①	②	③	④	⑤	⑥
FFFFH〜8000H	RAM 32Kバイト	RAM 32Kバイト	RAM 32Kバイト	RAM 32Kバイト	RAM 32Kバイト	RAM 32Kバイト
8000H〜0H	ROM 32Kバイト N80-BASIC	増設RAM 32Kバイト (バンク0)	増設RAM 32Kバイト (バンク1)	増設RAM 32Kバイト (バンク2)	増設RAM 32Kバイト (バンク3)	RAM 32Kバイト (バンク4)
I/OアドレスE2H OUT DATA	0,0,0,0,0,0,0,0	0,0,0,1,0,0,0,1	0,0,0,1,0,0,0,1	0,0,0,1,0,0,0,1	0,0,0,1,0,0,0,1	0,0,0,1,0,0,0,1
I/OアドレスE3H OUT DATA		0,0,0,0	0,0,0,1	0,0,1,0	0,0,1,1	0,1,0,0 *

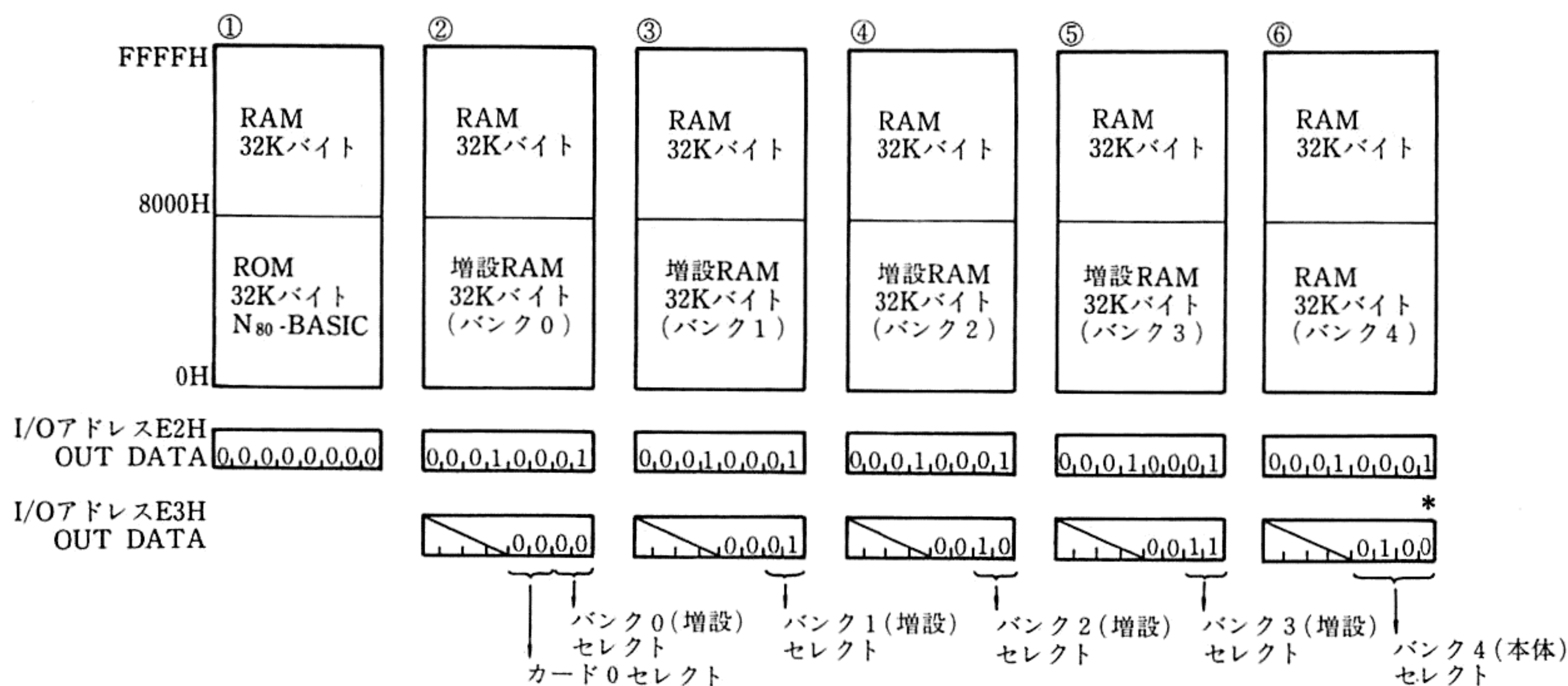



図12．8　増設RAMのメモリマップ2

＊ 前ページ(注　意)参照。

12.4 VRAM(Video RAM)

(1) テキストVRAM

MKII のテキストVRAMは，メインメモリ上のアドレスF300H～FFB7Hの約3K Byteに配置されています。テキストVRAMは，表示エリア(画面上の1つのキャラクタと1対1に対応)とアトリビュートエリア(画面の表示を修飾するためのもの)とによって構成されています。

図12．9に画面とテキストRAMとの対応を示します。

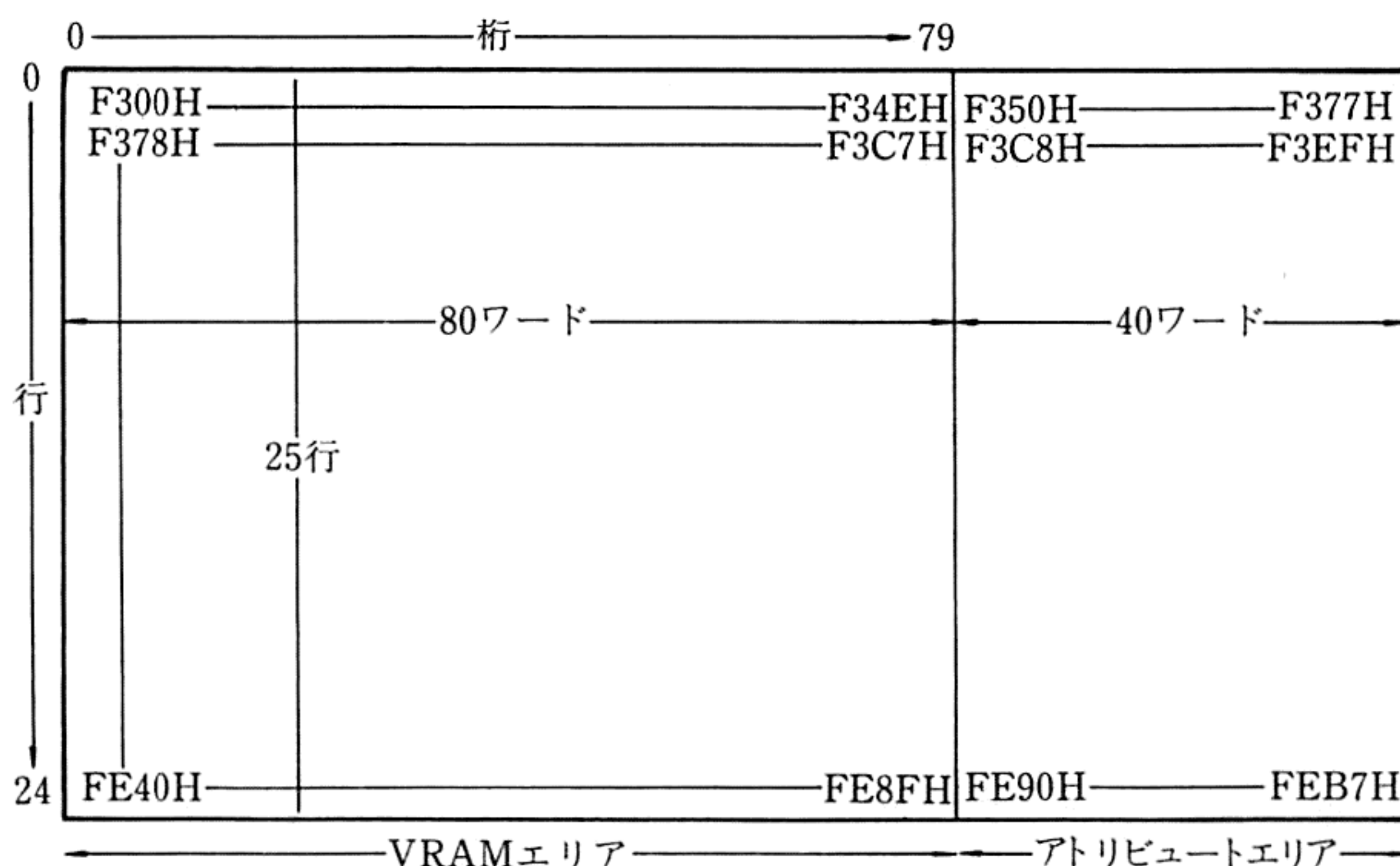



図12．9　テキストVRAMの構成

(2) グラフィックVRAM

MKIIのグラフィック VRAM は通常はCRT コントローラにより アクセスされていますが，バンク切り換えによりCPUがアクセスすることができます。

① 640×200 ピクセルグラフィックス

1ビットが1ピクセルに対応しています。図12.10 に画面とグラフィック VRAM との対応とグラフィックのピクセルとデータのビットとの対応を示します。

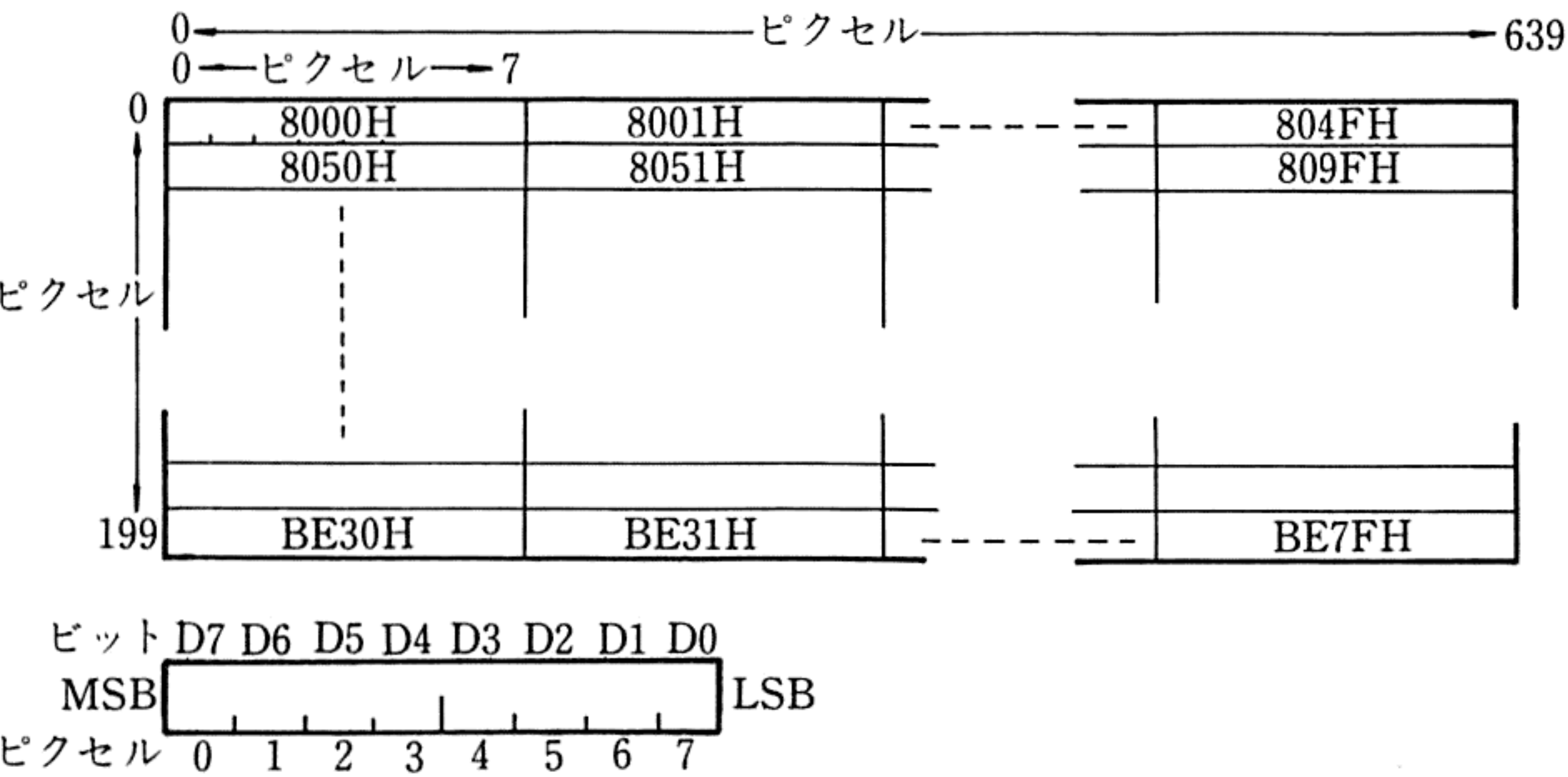



図12.10 グラフィックVRAMの構成(640×200ピクセルモード)

② 320×200 ピクセルグラフィックス

320×200 ピクセルのグラフィックスでは2ビットで1ピクセルが構成され，この2ビットにより4色が指定されます。

図 12.11 に画面とグラフィック VRAM との対応と，グラフィックのピクセルとデータのビットの対応を示します。

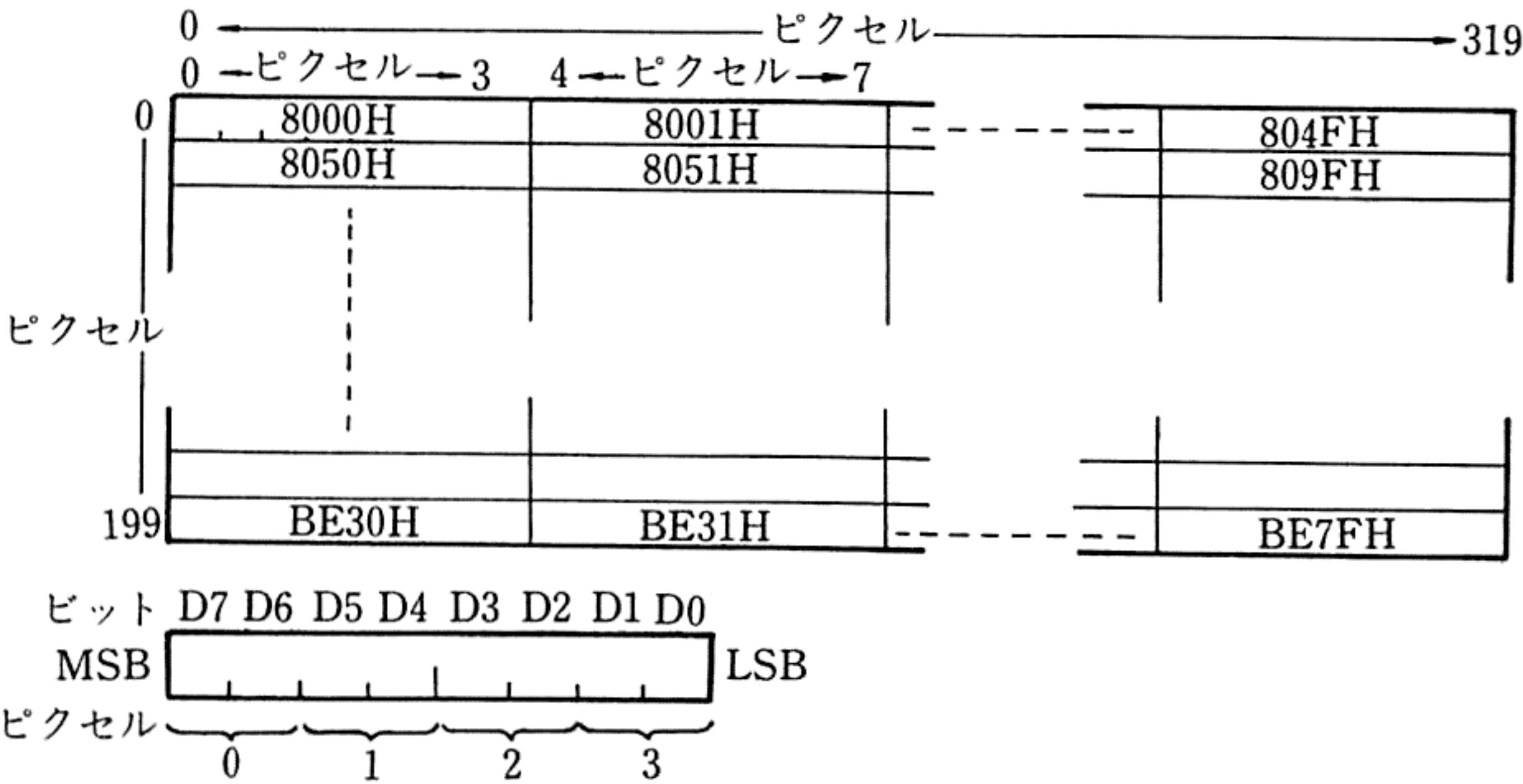



図12.11 グラフィックVRAMの構成(320×200ピクセルモード)

第13章
入出力インタフェース

MKII には，次に示す入出力インタフェースが標準装備されています。

カラー CRT インタフェース
白黒 CRT インタフェース
オーディオ・カセットインタフェース
RS-232C インタフェース
プリンタインタフェース
ミニフロッピィディスク用インタフェース
汎用 I/O ポート
漢字 ROM ボード用 I/O ポート
拡張用スロット
キーボード

以下にこれらのインタフェースの入出力規格を述べます。

13.1 カラーCRTインタフェース

MKII は専用カラーディスプレイに接続するためのインタフェースを内蔵しています。このコネクタからの出力信号は背面パネルにある8ピンのDIN コネクタに出力されています。

次に出力信号について説明します。

表13.1 カラーCRTインタフェース

端子番号	信　号　名		方向
1	VDD	+12V	出力
2	GND	信号グランド	
3	COLOR CLK	カラーバースト×4倍のクロック	出力
4	$\overline{\text{HSYNC}}$	水平同期信号	出力
5	$\overline{\text{VSYNC}}$	垂直同期信号	出力
6	R	ビデオ信号(Red)	出力
7	G	ビデオ信号(Green)	出力
8	B	ビデオ信号(Blue)	出力

8
7 6
3 1
5 4
2

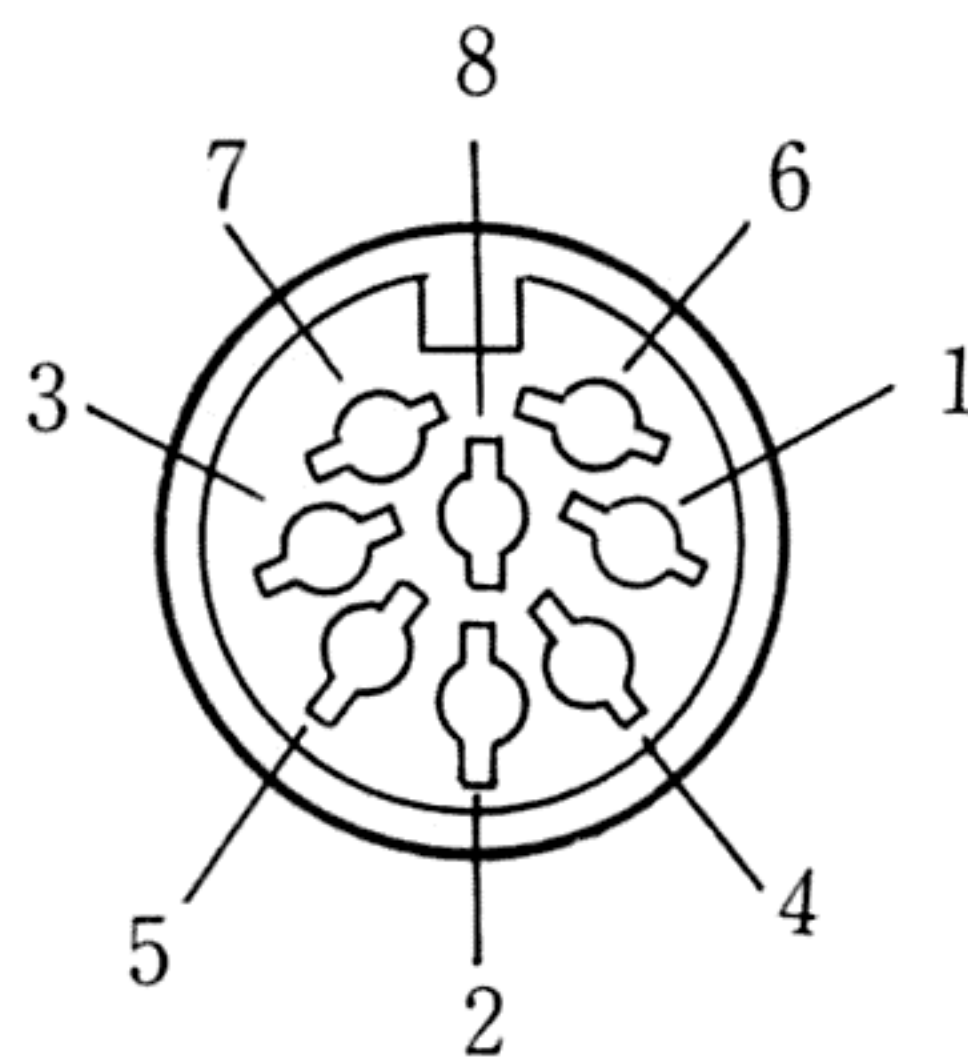



図13．1　カラーCRTコネクタピンコネクション

(1)　**VDD**

電源端子で　+12V　が出力されています。

これはオプションのカラーモジュレータに電源を供給するためのものです。

(2)　**GND**

信号グランドで，全ての信号に対してのグランド電位を与えるものです。

(3)　**COLOR CLK**

この信号も，オプションのカラーモジュレータを駆動するためのもので，TTL レベル，14.32MHz のクロックが出力されています。

(4)　**$\overline{\text{HSYNC}}$**

水平同期信号で，TTL レベル，負論理の信号が，出力されています。

(5)　**$\overline{\text{VSYNC}}$**

垂直同期信号で，TTL レベル，負論理の信号が，出力されています。

(6)　**R**

赤色のビデオ信号で，TTL レベル，正論理の信号が，出力されています。

(7)　**G**

緑色のビデオ信号で，TTL レベル，正論理の信号が，出力されています。

(8) **B**

青色のビデオ信号で，TTL レベル，正論理の信号が，出力されています。

13.2 白黒CRTインタフェース

MKII は白黒モニタテレビに接続するためのインタフェースを内蔵しています。このインタフェースからの出力信号は，背面パネルにある5ピンの DIN コネクタに出力されています。

次に出力信号について説明します。

表13. 2　白黒CRTインタフェース

端子番号	信　号　名		方向
1	VDD	+12V	出力
2	GND	信号グランド	
3	VIDEO OUT	ビデオ出力信号	出力
4	NC		
5	LPEN	ライトペン入力	入力

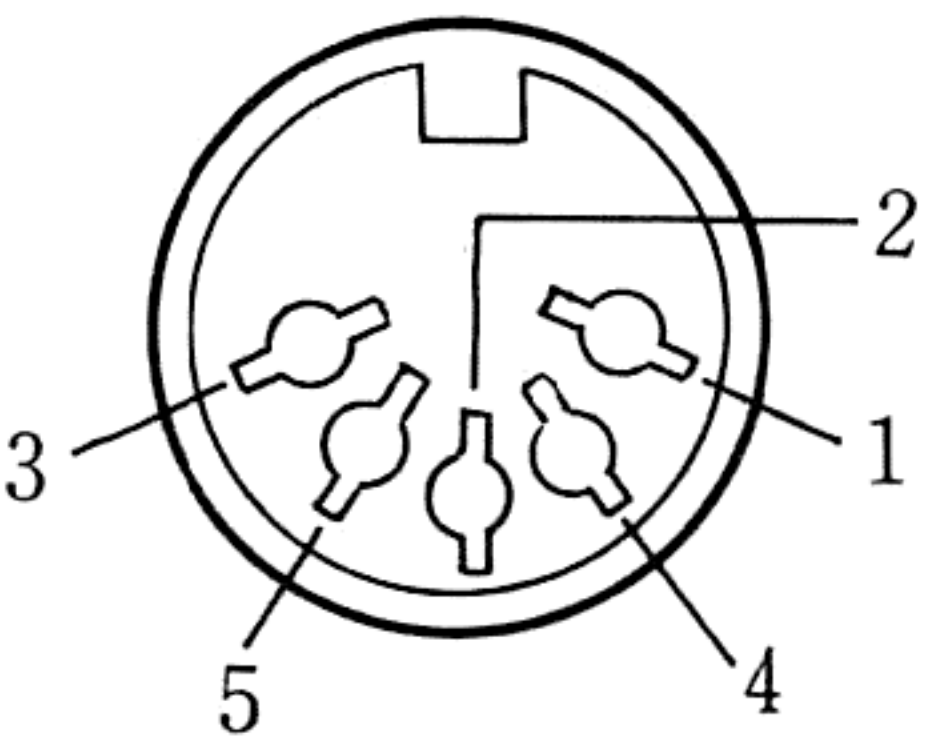



図13. 2　白黒CRTコネクタピンコネクション

(1) **VDD**

電源端子で，+12V が，出力されています。

(2) **GND**

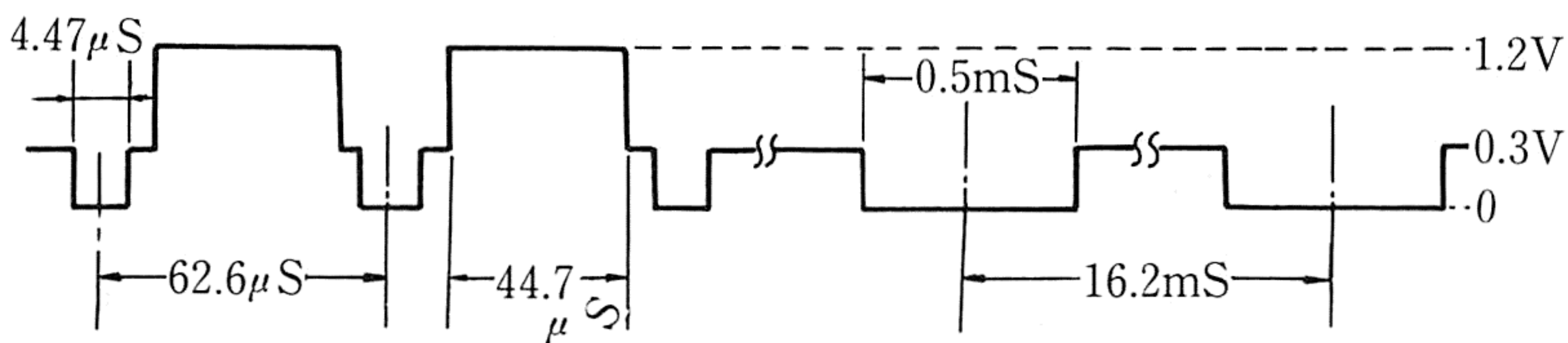
信号グランドで，全ての信号に対してのグランド電位を与えるものです。

(3) **VIDEO OUT**

ビデオ信号の出力端子です。この端子から出力されるビデオ信号は，コンポジットビデオ信号で映像信号と複合同期信号とが，すでに合成された信号です。

又，カラーモードで使用した場合，指定した色によって輝度変調がかかり，映像信号の DC レベルを色に応じて変えることができます。これによって，白黒テレビジョンを使用した場合でも，カラーモードにセットすることによって，画面に濃淡をつけることができます。

次に，ビデオ信号を示します。

同期パルスの正確なタイミング

(4) **LPEN**

ライトペンのストローブ信号入力端子です。

13.3 オーディオカセットインタフェース

MKII は，オーディオカセットへのデータの書き込み及び，オーディオカセットからのデータの読み込みを行うためのインタフェースを内蔵しています。このインタフェースの入出力信号は，背面パネルにある 8 ピンの DIN コネクタに接続されています。

次に，オーディオカセットへの書き込み方式について説明します。

① 書き込み方式

マーク 2400Hz，スペース 1200HzのFSK方式

② スピード

600bps

③ リモートコントロール機能

オーディオカセットテープレコーダについているリモート端子を使用することによって，テープの自動スタート／ストップが可能です。

注　意 オーディオカセットにデータを入出力するコマンドが，終了すると自動的にリモート信号がOFFとなりますので，テープの早送り巻もどしを行う場合は，リモート端子よりコネクタをぬくか **MOTOR** コマンドにより，リモート信号を ON にしてから行って下さい。

次に，入出力信号について説明します。

表13．3　オーディオカセットインタフェース

端子番号	信　号　名		方向
1	NC		
2	GND	信号グランド	
3	NC		
4	REC	オーディオカセット書き込み信号	出力
5	MON	オーディオカセット読み込み信号	入力
6	REM＋	リモート	出力
7	REM－	リモート	出力
8	NC		

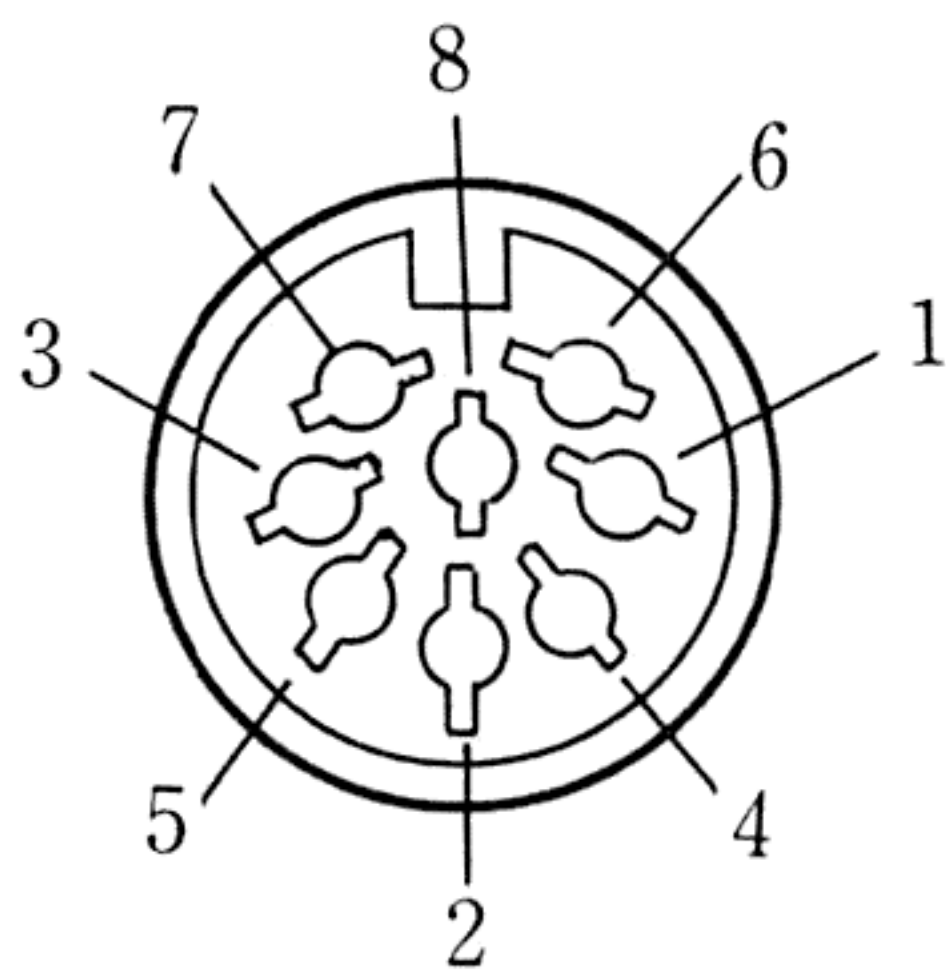



図13．3　オーディオカセットコネクタピンコネクション

(1) **GND**

信号グランドで，全ての信号に対してのグランド電位を与えるものです。

(2) **REC**

MKII から，オーディオカセットテープレコーダへの出力信号で，オーディオカセットテープレコーダのマイク端子へ接続します。なお，この信号は，付属ケーブルの赤色プラグに出力されています。

(3) **MON**

オーディオカセットテープレコーダからの入力信号で，オーディオカセットテープレコーダのイヤホーン端子に接続します。

なおこの信号は，付属ケーブルの白色プラグに接続されています。

(4) **REM＋，REM－**

オーディオカセットテープレコーダのリモート端子を制御する信号で，オーディオカセットテープレコーダのリモート端子に接続します。

この信号は，**MKII** 内に内蔵されているリレーの接点信号で，オーディオカセットにデータを入出力するコマンド及び **MOTOR** コマンドより制御されます。

13.4 RS-232Cインタフェース

MKII は，EIA (Electronic Industries Association) の定めたシリアルデータ転送規格の RS-232C インタフェースを内蔵しています。このインタフェースの入出力信号は背面のパネルにある 25ピンの Dサブコネクタに接続されています。

このインタフェースは **MKII** の ターミナルモードにより制御されます。

次に，入出力信号について説明します。

表13．4　RS-232Cインタフェース

端子番号	信　号　名		方向
1	GND	信号グランド	
2	$\overline{\text{TXD}}$	送信データ	出力
3	$\overline{\text{RXD}}$	受信データ	入力
4	RTS	送信要求	出力
5	CTS	送信可	入力
6	DSR	データセットレディ	入力
7	GND	信号グランド	
8	DCD	キャリア検出信号	入力
9	NC		
10	NC		
11	NC		
12	NC		
13	NC		
14	NC		
15	NC		
16	NC		
17	NC		
18	NC		
19	NC		
20	DTR	データ端末レディ	出力
21	NC		
22	NC		
23	NC		
24	NC		
25	NC		

NCは，No Connectionの略です。

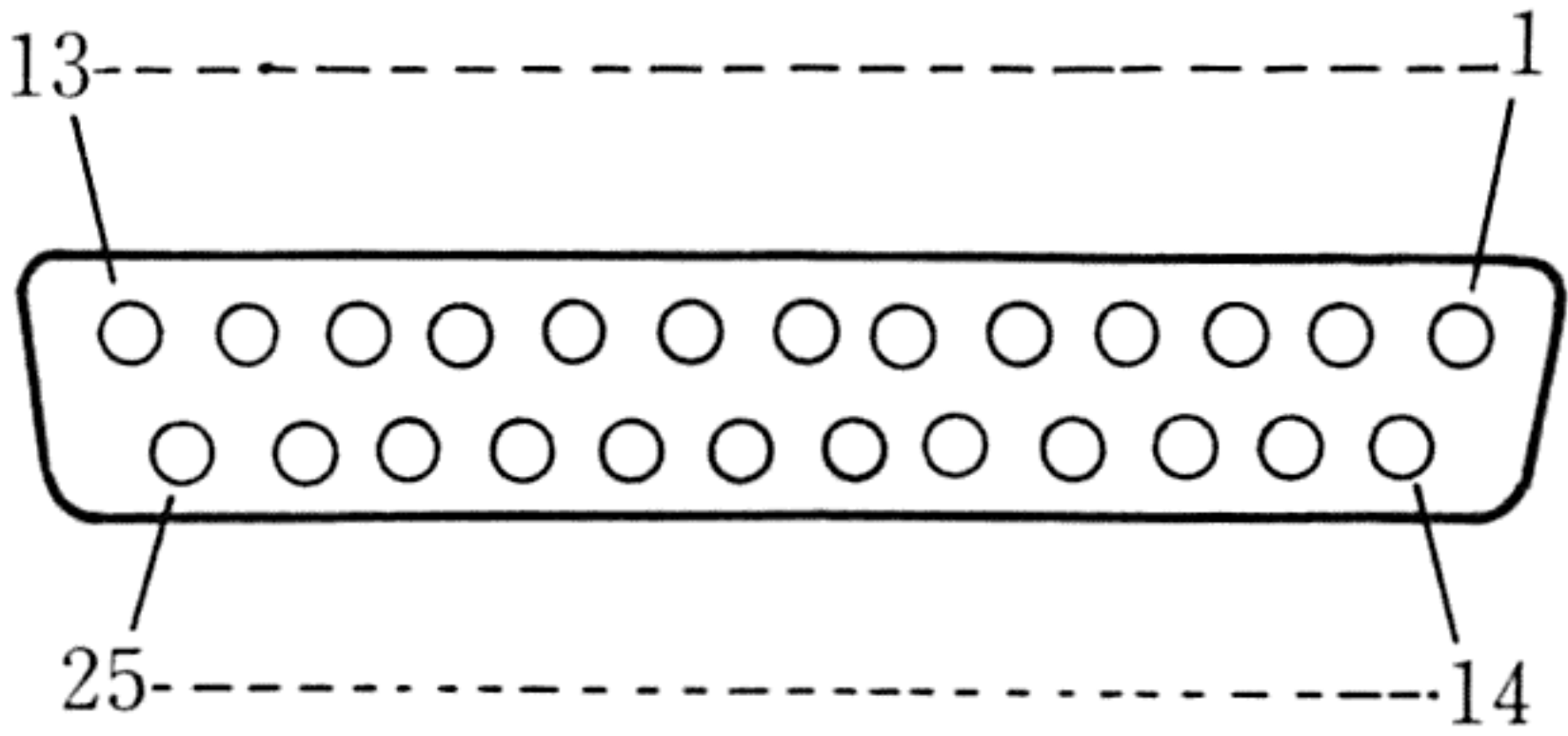



図13．4　RS-232Cコネクタピンコネクション

(1) **GND**

信号グランドで，全ての信号に対してグランド電位を与えるものです。

(2) **$\overline{\text{TXD}}$**

送信データで，**MKII**から外部シリアル入出力機器へ，シリアルデータを送出するためのものです。

(3) **$\overline{\text{RXD}}$**

受信データで，外部シリアル入出力機器から，**MKII**へシリアルデータを入力するためのものです。

(4) **RTS**

送信要求信号で，モデムの送信機能を制御します。

(5) **CTS**

外部シリアル入出力機器からの入力信号で，外部シリアル入出力機器が，データの送信可能状態にあるか否かを**MKII**に示す信号です。

(6) **DSR**

外部シリアル入出力機器からの入力信号で，外部シリアル入出力機器が，データの送受信ができる状態にあるか否かを**MKII**に示す信号です。

(7) **DCD**

外部シリアル入出力機器からの入力信号で，受信キャリアが，規格内にあるか否かを**MKII**に示す信号です。

(8) **DTR**

外部シリアル入出力機器への出力信号で，**MKII**側の送受信機能が，レディになったことを外部シリアル入出力機器へ示す信号です。

(9) **ボーレートの指定**

RS-232C のボーレートは背面パネルにあるジャンパスイッチにより切換えることができます。

次にジャンパスイッチの位置とボーレートの関係を示します。

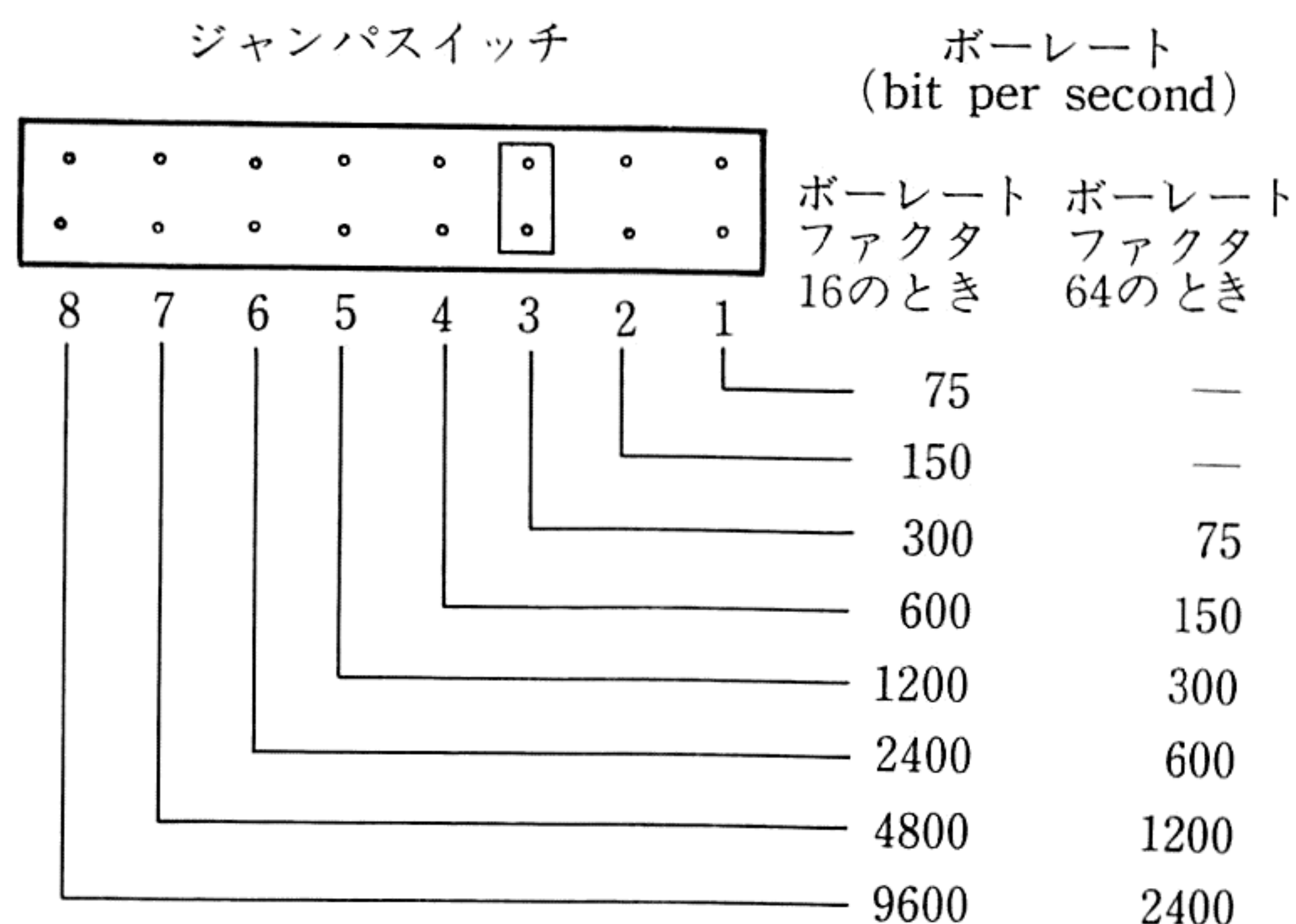



図13.5　ジャンパスイッチとボーレート

13.5 プリンタインタフェース

MKII はプリンタ用パラレルインタフェース（セントロニクス社製プリンタ仕様に準ずる）を内蔵しています。このインタフェースの入出力信号は　背面パネルの14ピンアンフェノールコネクタに接続されています。

次に入出力信号について説明します。

表13.5　プリンタインタフェース

端子番号	信号名
1	$\overline{\text{PSTB}}$
2	PDB0
3	PDB1
4	PDB2
5	PDB3
6	PDB4
7	PDB5
8	PDB6
9	PDB7
10	NC
11	BUSY
12	NC
13	NC
14	GND

ピンコネクション

7------------1

14-------------8

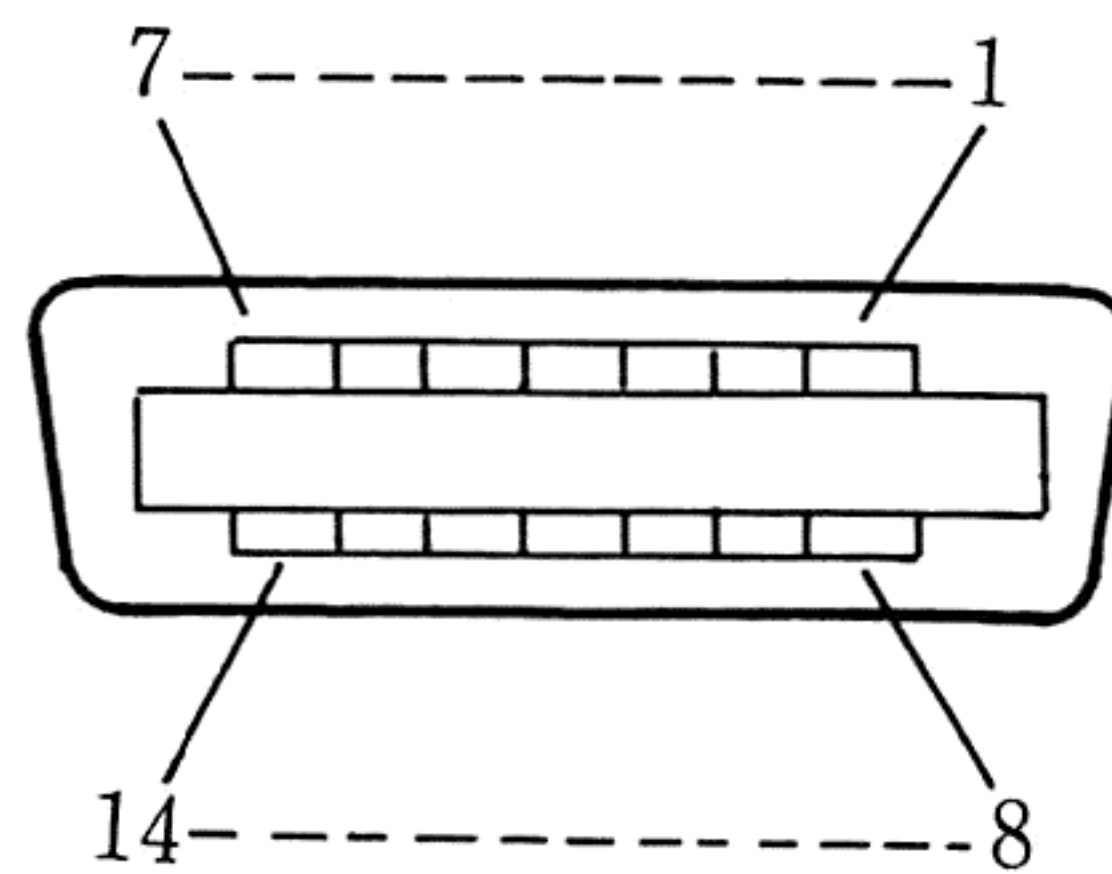



図13.6　プリンタコネクタピンコネクション

(1)　$\overline{\textbf{PSTB}}$

プリンタに出力するライトストローブ信号で，プリンタにデータを出力する時のライト信号として使います。

(2)　**PDB0～PDB7**

プリンタに出力する8ビットのパラレルデータバスです。

(3)　**BUSY**

プリンタからのレディ信号を受信します。

この信号が，LOW レベルのとき **MKII**は，プリンタにデータを出力することができます。

(4)　**GND**

信号グランドで全ての信号に対してグランド電位を与えるものです。

すべての入出力は，TTL レベルで行います。

13.6 ミニフロッピィディスクインタフェース

MKIIはミニフロッピィディスク用パラレルインタフェースを内蔵しています。このインタフェースの入出力信号は背面パネルの36ピンアンフェノールコネクタに接続されています。このインタフェースには **PC-8031** または **PC-8031-2W** などPC-8000シリーズのインテリジェントタイプのミニフロッピィディスクユニットを接続することができます。

表13．6　ミニフロッピィディスクインタフェース

端子番号	信 号 名	端子番号	信 号 名
1	PB0	19	PA0
2	PB1	20	PA1
3	PB2	21	PA2
4	PB3	22	PA3
5	PB4	23	PA4
6	PB5	24	PA5
7	PB6	25	PA6
8	PB7	26	PA7
9	$\overline{\text{EXTON}}$	27	PC4
10	GND	28	PC5
11	GND	29	PC6
12	GND	30	PC7
13	GND	31	PC0
14	GND	32	PC1
15	GND	33	PC2
16	GND	34	PC3
17	GND	35	$\overline{\text{RESET}}$
18	GND	36	GND

18------------------1

36------------------19

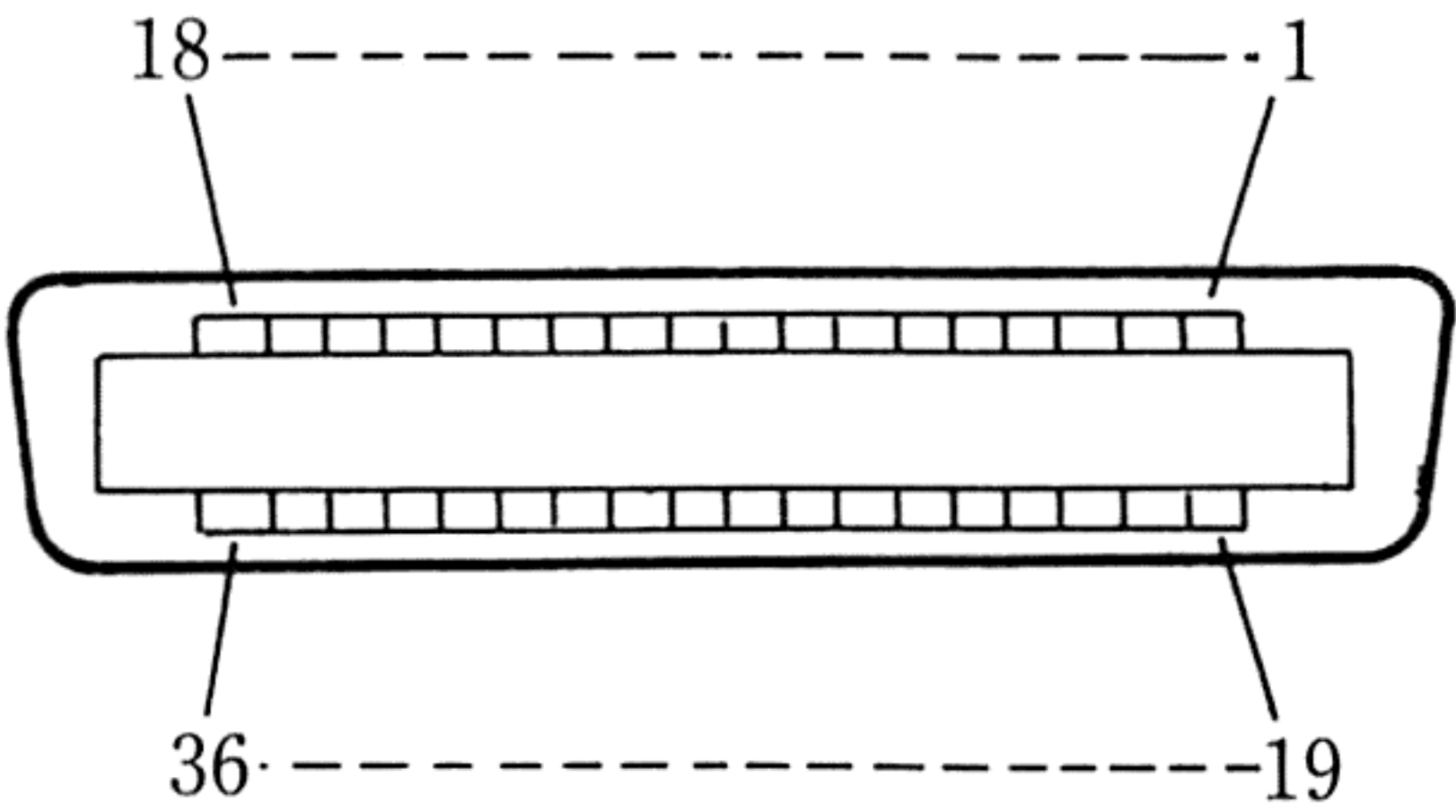



図13．7　ミニフロッピィディスクコネクタピンコネクション

13.7　汎用I/Oポート

MKII は汎用の I/O ポートを持っています。 この I/O ポートの入

出力信号は背面パネルの９ピンのＤサブコネクタに接続されています。このポートは入力信号2本，出力信号3本，割込み1chが接続されています。**MKⅡ**の**N80-BASIC**ではサポートされておりませんが多様なアプリケーションに対応することができるでしょう。

表13．7　汎用I/Oポートインタフェース

端子番号	信　号　名
1	UIP2
2	UIP1
3	UOP1
4	UOP2
5	GND
6	UOP3
7	GND
8	$\overline{\text{UINT2}}$
9	+5 V

1----------5

6----------9

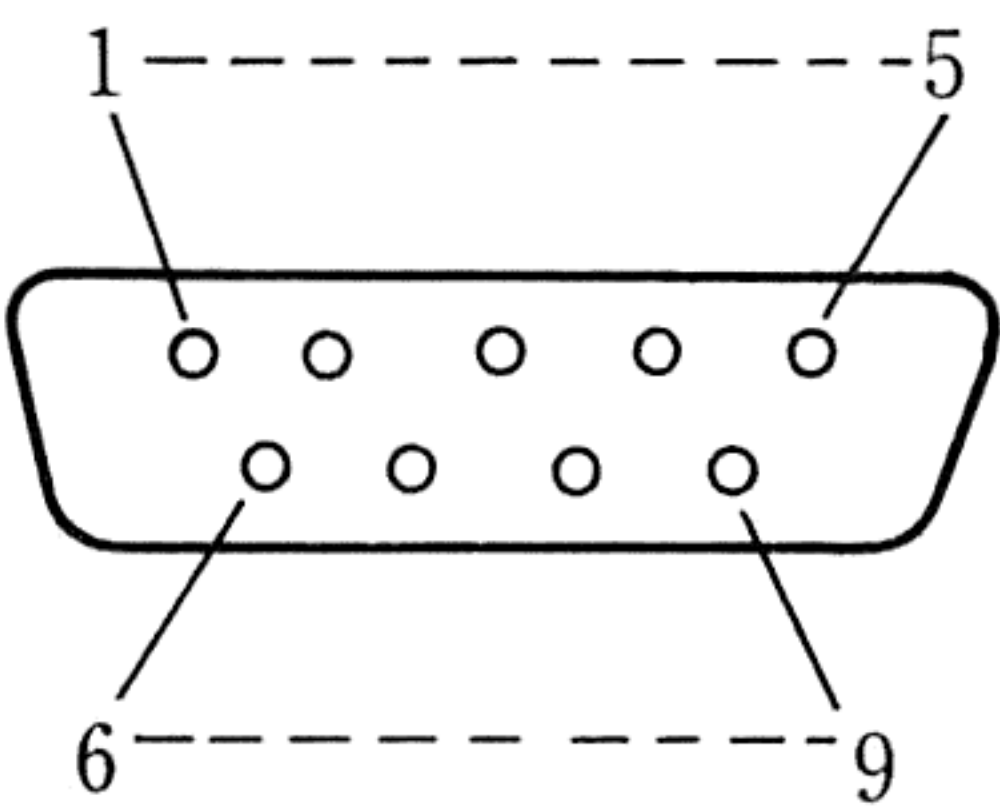



図13．8　汎用I/Oポートコネクタピンコネクション

13.8 漢字ROMボード用I/Oポート

MKⅡの**N80-BASIC**は漢字を表示するための命令，**CMD PUT**を持っています。この漢字命令を使用するためにはオプションの漢字ROMボード(**PC-8001MKⅡ-01**)を本体基板内のI/O BUSに接続する必要があります。取り付け方法は漢字ROMボードの説明書を参照してください。

表13．8　漢字ROMボードインタフェース

端子番号	信　号　名	端子番号	信　号　名
1	RDB0	13	RDB4
2	RDB1	14	RDB5
3	RDB2	15	RDB6
4	RDB3	16	RDB7
5	RAB0	17	RAB4
6	RAB1	18	RAB5
7	RAB2	19	RAB6
8	RAB3	20	RAB7
9	$\overline{\mathrm{RIOR}}$	21	$\overline{\mathrm{RIOW}}$
10	+5V	22	GND
11	+5V	23	GND
12	+5V	24	GND

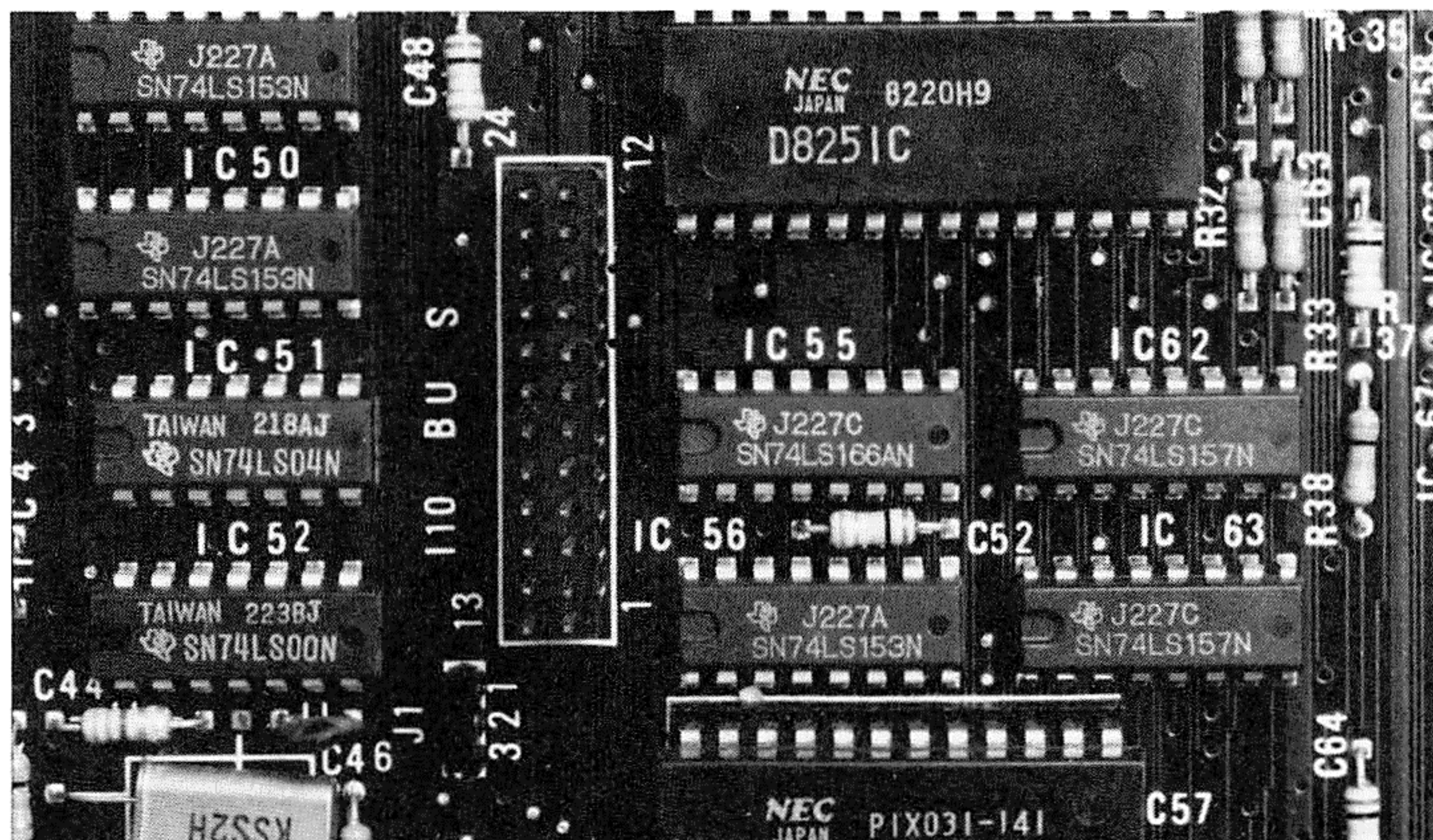



図13．9　漢字ROMボードコネクタ

13.9 拡張用スロットバス

MKII には，拡張用スロットとして，本体後部に2スロット内蔵しています。この拡張用スロットには，8インチ標準フロッピィディスク用インターフェースボードや音声認識ボードなどを実装することができます。ま

た，個人で設計した回路を接続することもできますが，その場合は，**PC-8012-01**ユニバーサルボードを使用してください。それ以外のボードですと，サイズやピン間隔が合わない場合などは，故障の原因となりますから注意してください。

(注　意) スロット #1 とスロット #2 は信号が一部異なっています。8インチ標準フロッピィディスク用インタフェースボードは，スロット #2に実装してください。拡張 RAM ボードなどは，どのスロットでも使用できます。

MKIIの拡張スロットバスは **PC-8801**, **PC-8012** のスロットバスと上位コンパチブルです。詳細は，拡張スロットバスのピン配置図を参照してください。

B36 ----------------- B1

A36 ----------------- A1

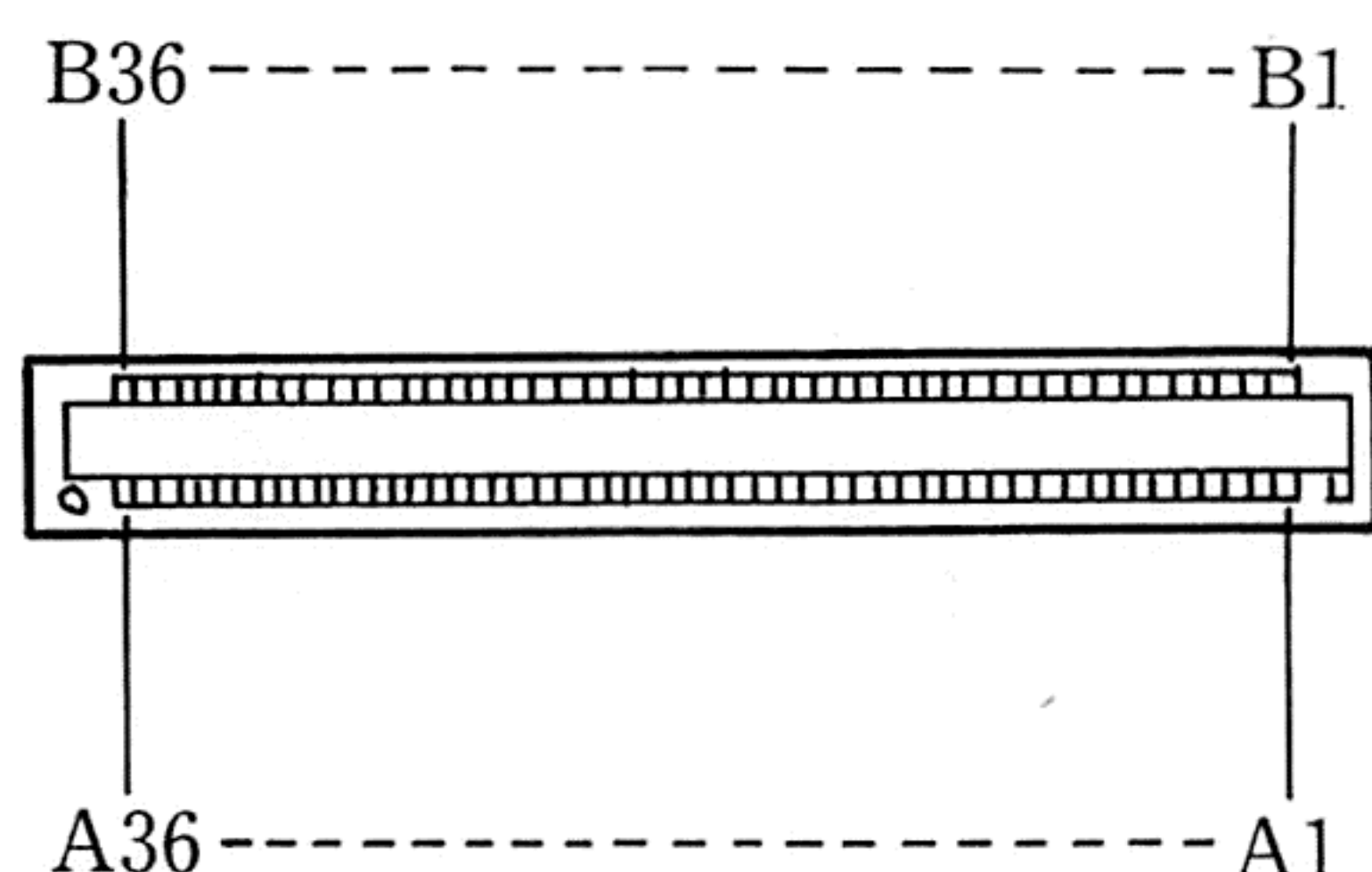



図13.10　拡張用スロットピンコネクション

(注　意) $\overline{\text{FDINT1}}$と$\overline{\text{INT1}}$は内部でワイヤード OR されていますので，8インチ標準フロッピィディスク用インタフェースボードを接続したときは，$\overline{\text{INT1}}$は使用しないでください。

また$\overline{\text{FDINT2}}$と$\overline{\text{INT0}}$も内部でワイヤード OR されていますので FDINT2 を使用した場合は $\overline{\text{INT0}}$ は使用しないでください。

汎用 I/O ポートの割込み信号 $\overline{\text{UINT2}}$ はスロットバスの $\overline{\text{INT2}}$ と同じ割り込みチャンネルを使用しています。

表13．9　拡張用スロットバスインタフェース(その１)

端子番号	MKIIバス１		MKIIバス２	
	信号名(サイドA)	信号名(サイドB)	信号名(サイドA)	信号名(サイドB)
1	GND	GND		
2	GND	GND		
3	+5	+5		
4	+5	+5		
5	AB0	$\overline{\mathrm{INT0}}$		
6	AB1	$\overline{\mathrm{INT1}}$		
7	AB2	$\overline{\mathrm{INT2}}$		
8	AB3	$\overline{\mathrm{INT3}}$		
9	AB4	$\overline{\mathrm{INT4}}$		
10	AB5	$\overline{\mathrm{MWAIT}}$		
11	AB6	$\overline{\mathrm{FDINT1}}$		
12	AB7	$\overline{\mathrm{FDINT2}}$		
13	AB8	DB0		
14	AB9	DB1		
15	AB10	DB2		
16	AB11	DB3		
17	AB12	DB4		
18	AB13	DB5		
19	AB14	DB6		
20	AB15	DB7		
21	$\overline{\mathrm{RD}}$	$\overline{\mathrm{MEMR}}$		
22	$\overline{\mathrm{WR}}$	POWER		
23	$\overline{\mathrm{MREQ}}$	$\overline{\mathrm{IOW}}$		
24	$\overline{\mathrm{IORQ}}$	$\overline{\mathrm{IOR}}$		
25	$\overline{\mathrm{M1}}$	$\overline{\mathrm{MEMW}}$		
26	$\overline{\mathrm{RAS0}}$	$\overline{\mathrm{DMATC}}$		
27	$\overline{\mathrm{RAS1}}$	FDRDY		
28	$\overline{\mathrm{RFSH}}$	$\overline{\mathrm{DRQ1}}$		$\overline{\mathrm{DRQ2}}$
29	$\overline{\mathrm{MUX}}$	$\overline{\mathrm{DACK1}}$		$\overline{\mathrm{DACK2}}$
30	$\overline{\mathrm{WE}}$	4CLK		
31	$\overline{\mathrm{ROMKILL}}$	$\overline{\mathrm{NMI}}$		
32	$\overline{\mathrm{RESET}}$	$\overline{\mathrm{WAITRQ}}$		
33	SCLK	+12		
34	CLK	−12		
35	V1	V1		
36	V2	V2		

空白の部分はバス１と同じ信号名が入ります。

表13.9　拡張用スロットバスインタフェース(その２)

端子番号	PC-8801のバス		PC-8012のバス	
	信号名 (サイドA)	信号名 (サイドB)	信号名 (サイドA)	信号名 (サイドB)
1				
2				
3				
4				
5		空き		
6		空き		
7		$\overline{\text{MWAIT}}$		
8		$\overline{\text{INT4}}$		
9		$\overline{\text{INT3}}$		
10		$\overline{\text{INT2}}$		$\overline{\text{INT5}}$
11				$\overline{\text{INT6}}$
12				$\overline{\text{INT7}}$
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				$\overline{\text{IOWR}}$
23				
24				
25				空き
26				空き
27				空き
28				空き
29				空き
30				空き
31				
32				$\overline{\text{WAIT}}$
33				
34				
35				
36				

空白の部分はMKⅡのバス１と同じ信号名が入ります。

拡張ボードの実装方法

MKII の後部にはスロットバスが2スロット有ります。ここには各種拡張ボードが入りますが，どのスロットも実装方法は同じです。(ただしスロット1とスロット2では信号が一部異なります) ここでは，例としてスロット1にボードを実装する手順を説明します。

① 作業を行なう前にコンセントから **MK II** の電源プラグを抜いてください。

② スロットバスのフタをはずします。

・フタを止めている2本のビスをはずすとフタは簡単にはずれます。

・ドライバは大きさの合ったものを使用してください。大きさが合わないものを使用すると，ネジ山をつぶす原因となります。

2本のビスをはずす

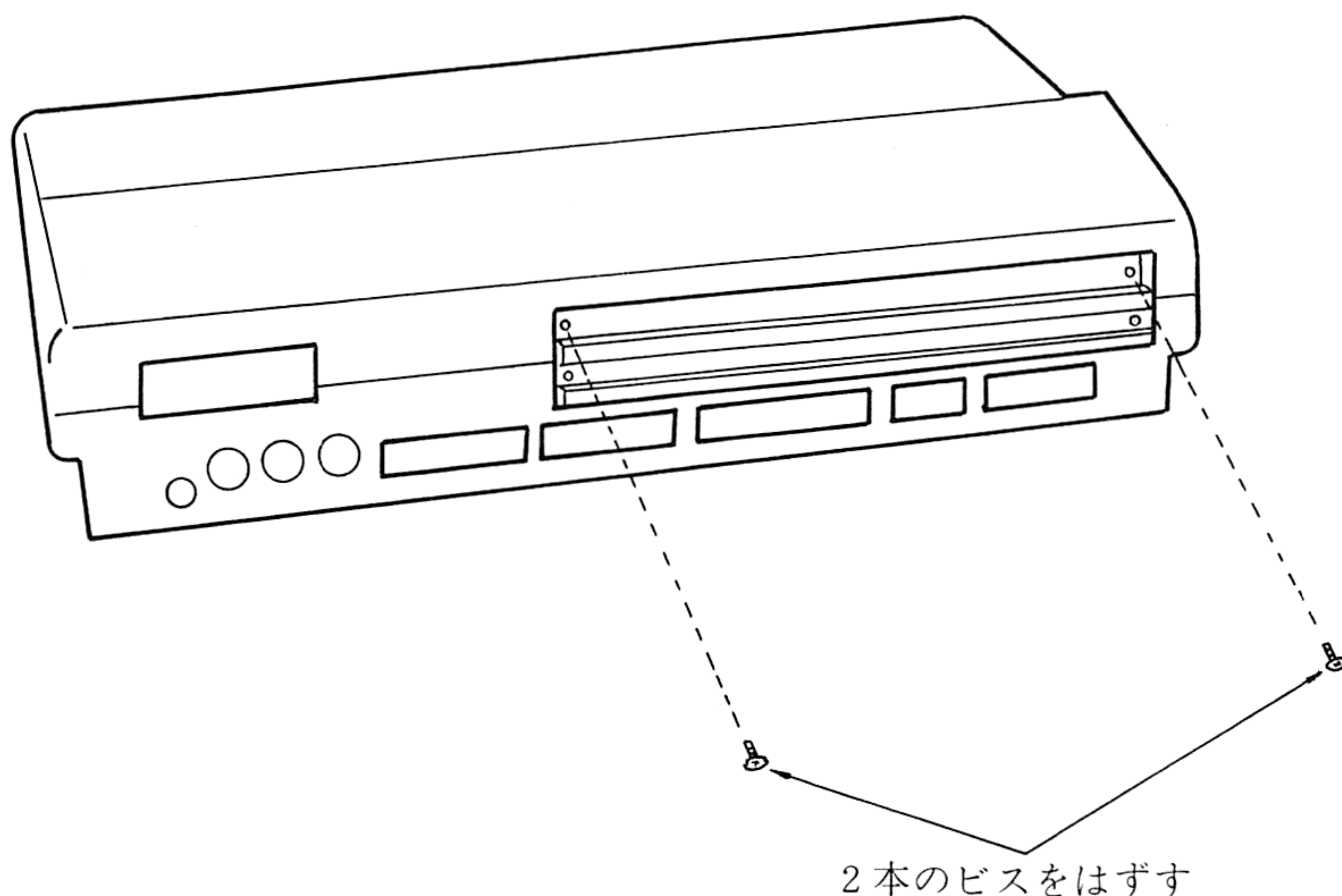



図13.11 スロットバスのフタをはずす

③ 拡張ボードを差し込みます。

・部品面（ IC などの部品が付いている面）を上にして，カードガ

イドのミゾにボードを合わせて差し込んでください。

・ボードが本体内部にほぼ隠れるところまでは軽く挿入できます。

カードガイドのミゾ

(部品面)

カード・プラ

この部分を押す

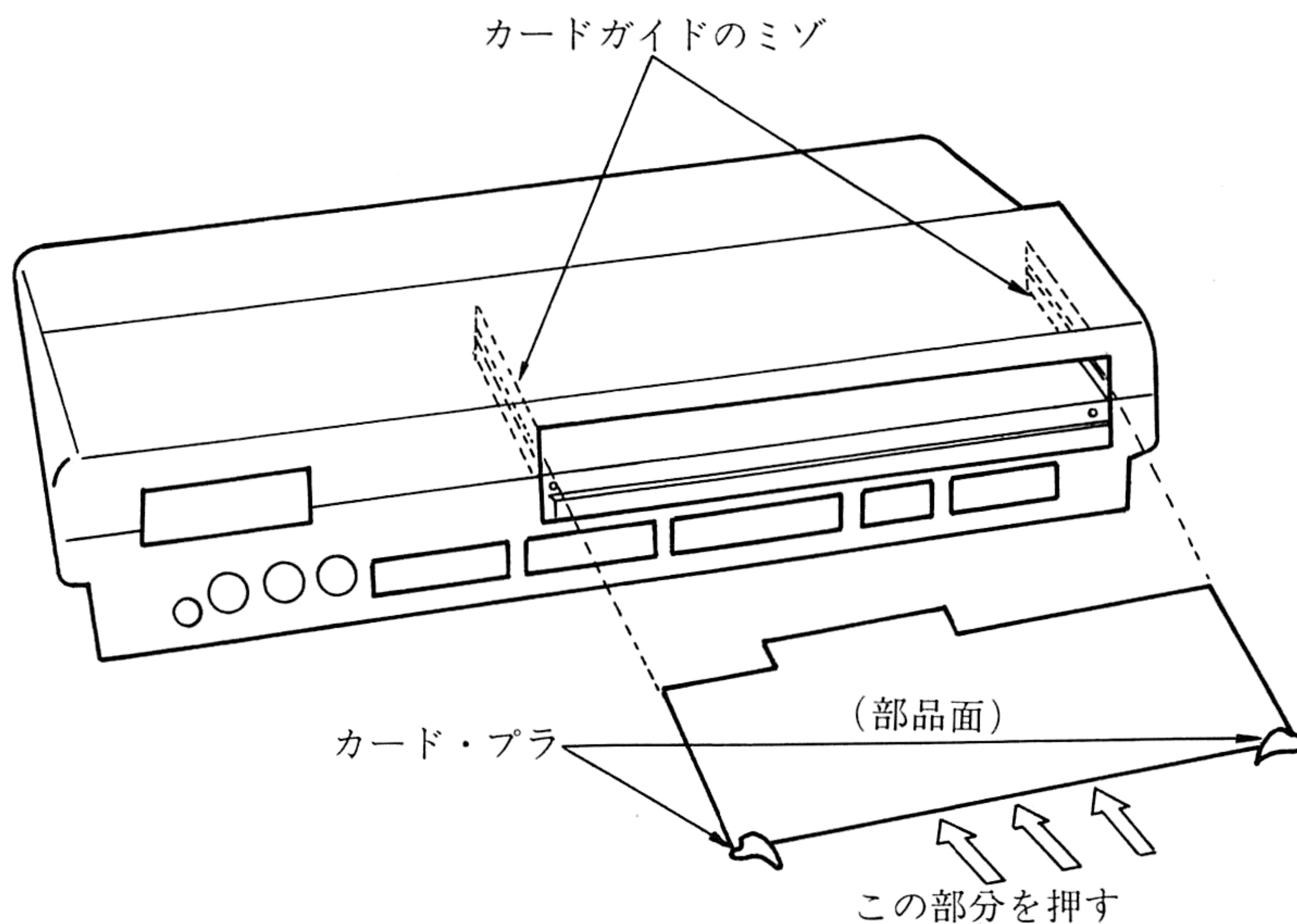



図13.12　拡張ボードを差し込む

・最後にカツンとショックがあるまで強く押し込みます。

・カードプラは押さないで下さい。破損の原因となります。

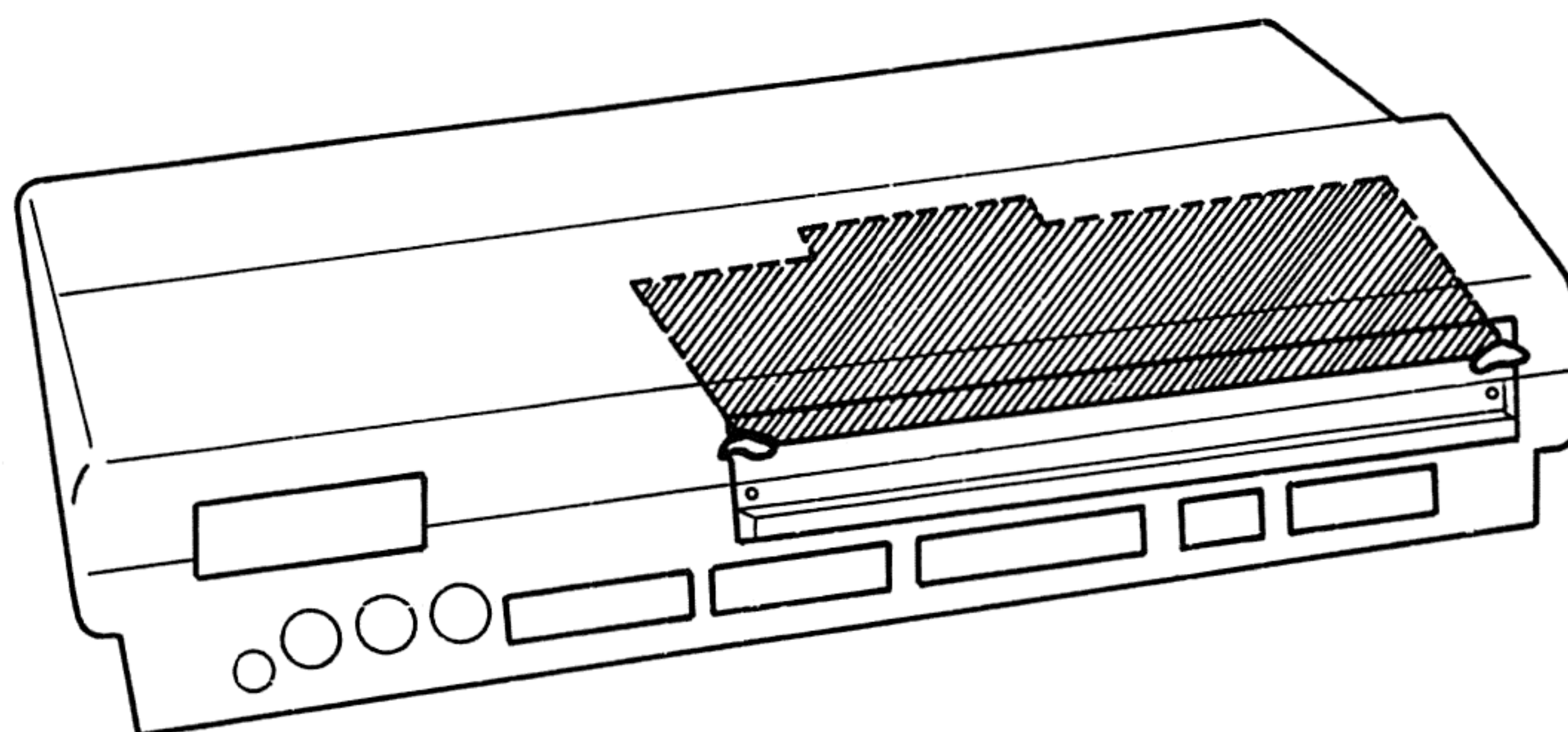



図13.13　ボードが実装されたところ

・ボードを軽く引張ってみて，抜けないことを確認してください。

④　スロットバスのフタを閉めます。

・実装したボードの上から②で外したフタを閉め，ビスで止めてください。

拡張ボードの取り外し方

①　実装のときと同じ手順でフタをはずしてください。

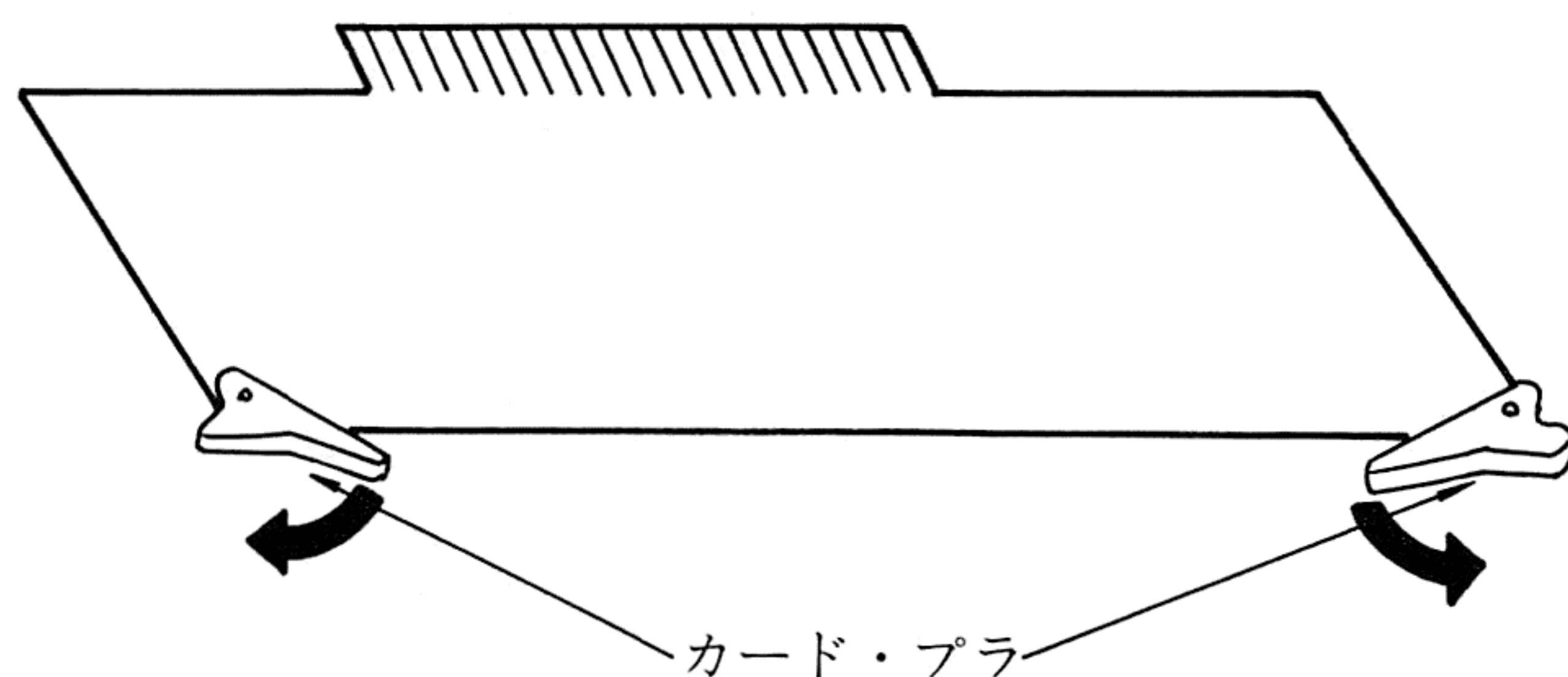
②　拡張ボードにはカードプラが付いていますので，これを図の矢印の方向へ動かすとボードは容易にはずれます。(一部カードプラが付いていない拡張ボードもあります)

カード・プラ

図13.14　カード・プラ

③　フタをして，ビスを止めてください。

13.10 キーボード

MKII のキーボードは，ソフトウェアセンス方式をとっています。マトリクス状に配置されたキースイッチは，CPU のインプット命令によりスキャンされ，押されたキーの情報は，ソフトウエアによって JIS コードに変換されます。

たとえば inp(&H09) の値が 01111111 (2進数)なら，**ESC** キーが押されています。

次に，キーボードの配列を示します。

データー

I/Oアドレス	LSB D0	D1	D2	D3	D4	D5	D6	D7 MSB
00H	0	1	2	3	4	5	6	7
01H	8	9	*	+	=	,	.	RETURN
02H	@ ゛	A チ	B コ	C ソ	D シ	E イ	F ハ	G キ
03H	H ク	I ニ	J マ	K ノ	L リ	M モ	N ミ	O ラ
04H	P セ	Q タ	R ス	S ト	T カ	U ナ	V ヒ	W テ
05H	X サ	Y ン	Z ッ ツ	[。「	¥ _	] ム 」	^ ヘ	− = ホ
06H	0 ヲ ワ	1 ! ヌ	2 " フ	3 # ァ ア	4 $ ゥ ウ	5 % ェ エ	6 & ォ オ	7 ▼ ャ ヤ
07H	8 (ュ ユ	9) ョ ヨ	: * ケ	; + レ	, < 、 ネ	. > 。 ル	/ ? ・ メ	− ロ
08H	HOME CLR	↑	→	INS DEL	GRPH	カナ	SHIFT	CTRL
09H	STOP	f・1	f・2	f・3	f・4	f・5	SPACE	ESC

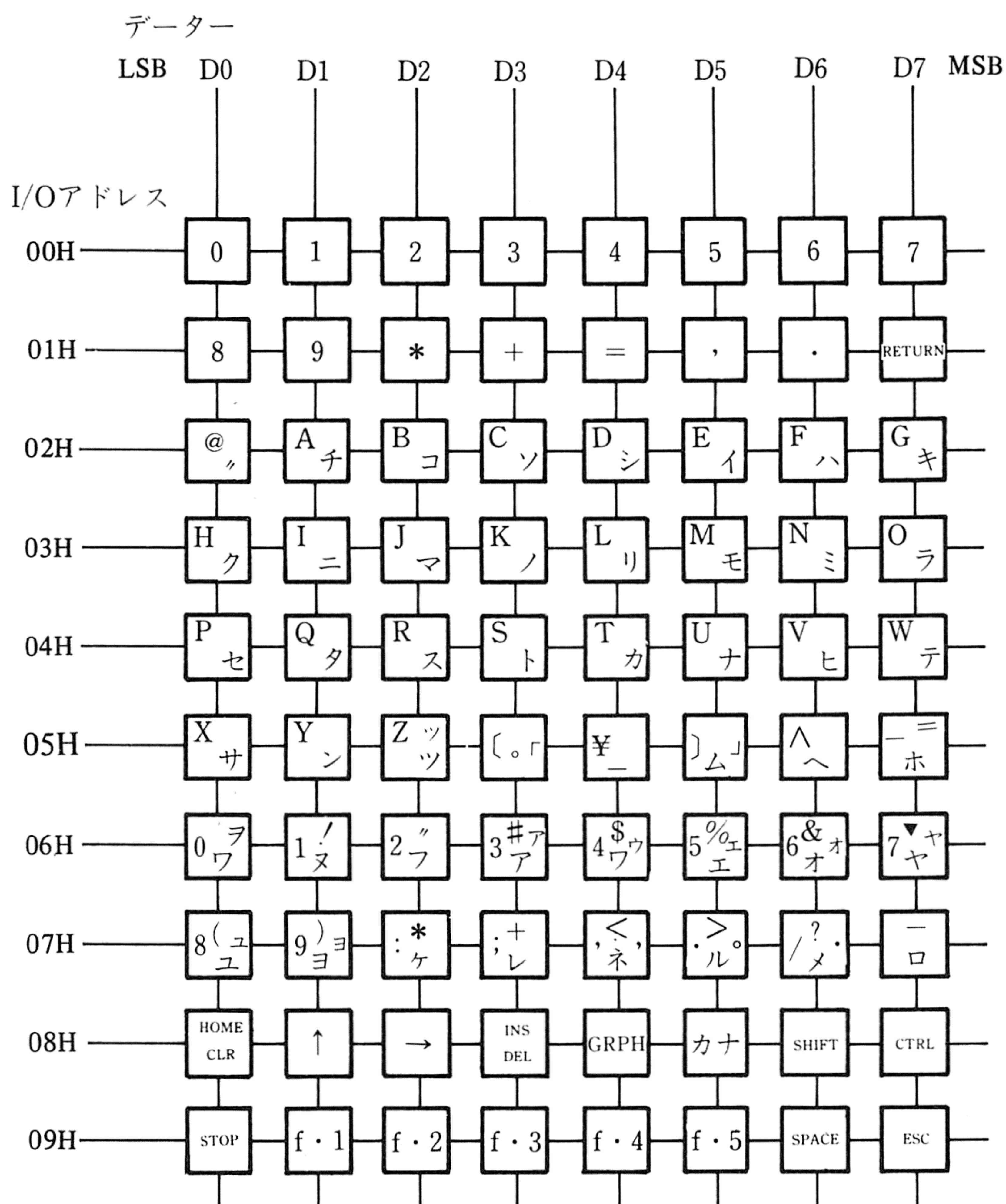



図13.15 キーボードマトリクス(その1)

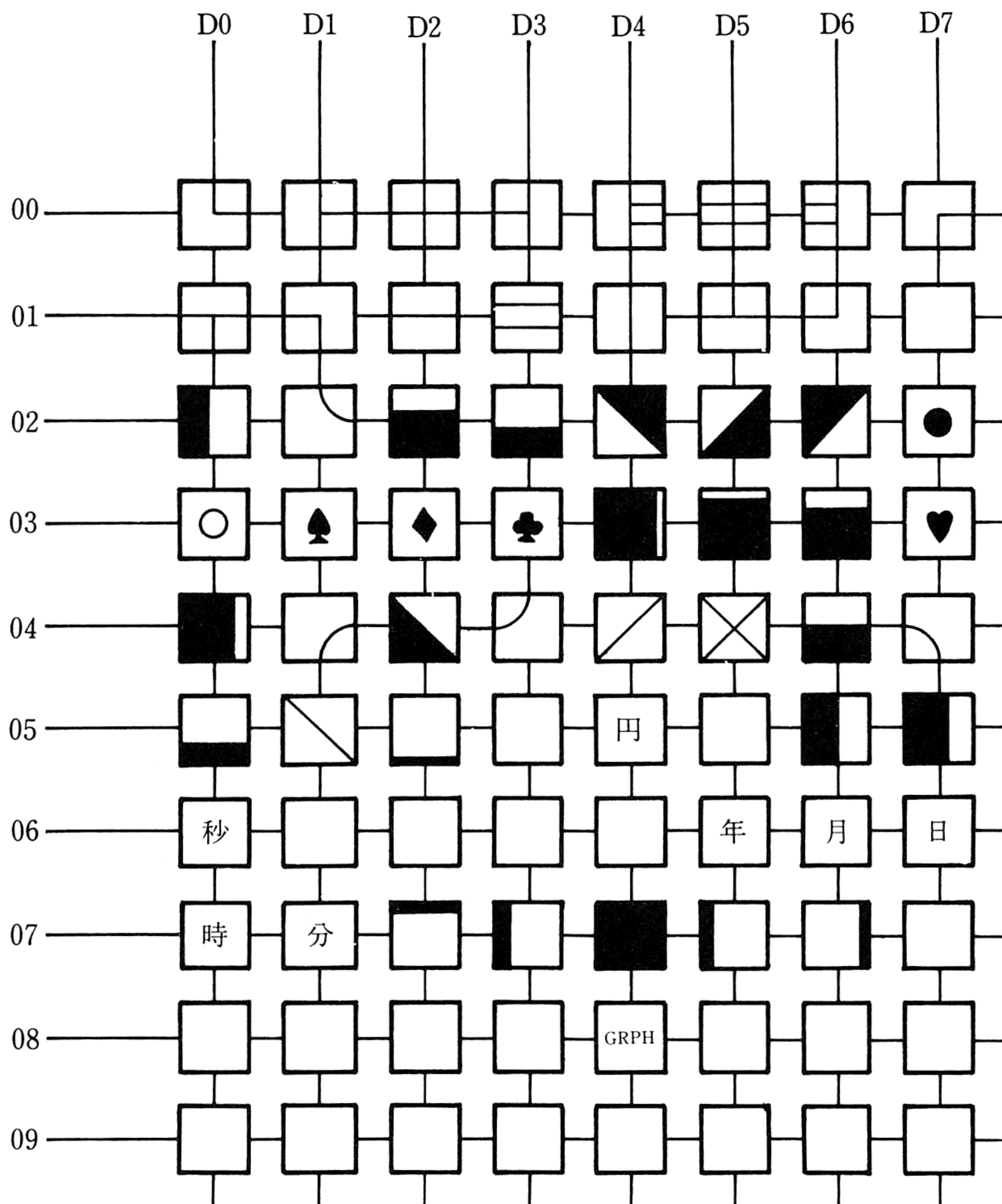



図13.16　キーボードマトリクス(その2)

第14章
割り込み

MKIIには，**割り込みの制御ロジック**が標準装備されています。この割り込み機能は **CPU**(μ**PD780C-1**)のモード２にシステムプログラム(**N80-BASIC**)により定義されています。

割込みチャネルは，**8レベルプライオリティ付**のものですが，その内システムで３レベル，FDC のリザーブとして２レベル，残りの **3レベル**がユーザーに開放されています。

14.1 μPD780C-1モード2割り込みについて

μPD780 には，ソフトウエアマスカブルな割り込みと，ノンマスカブルな割り込みの２種類があります。**MKII** ではそのどちらも使うことができますが，**N80-BASIC** が動作中は，マスカブル割り込みしか使用することができません。

マスカブル割り込み（以下 INT と記述します。）は，プログラムによって受け付けを禁止したり，許可したりすることができます。このことは，ユーザが中断されてはいけないタイミングの制約を持ったプログラムを走らせるときには，割り込みを禁止できることを意味しています。

マスカブル割り込みにはマスクフラグ IFF があり，プログラムによりセット / リセットすることにより，割り込みを禁止 / 許可することができます。

又，μPD780 には，割り込みレスポンスモードとしてモード０，モード１，モード２の３種類があり，プログラムによってモードの指定を行なった後，使用します。

MKⅡの **N80-BASIC** では，モード２が指定されています。 μPD780のモード２割り込みのレスポンスは，次のようになります。このモードでは，ユーザはすべての割り込みサービスルーチンに対して16ビットスタートアドレスのテーブルを整えます。

このテーブルは，メモリのどこにでも置くことができます。

割り込みが受付けられると，テーブルから希望の割り込みサービスルーチンのスタートアドレスを得るために，16 ビットのポインタが必要となります。

このポインタの上位８ビットは，Ｉレジスタから得られます。

ポインタの下位８ビットは，割り込み制御ロジックより割り込みアクノレッジサイクル時に CPU に与えなければなりません。

下位アドレス

割り込みサービスルーチンスターティングアドレステーブル

下位アドレス
上位アドレス

上位アドレス

Ｉレジスタ

割り込み制御ロジックよりのテーブルポインタデータ

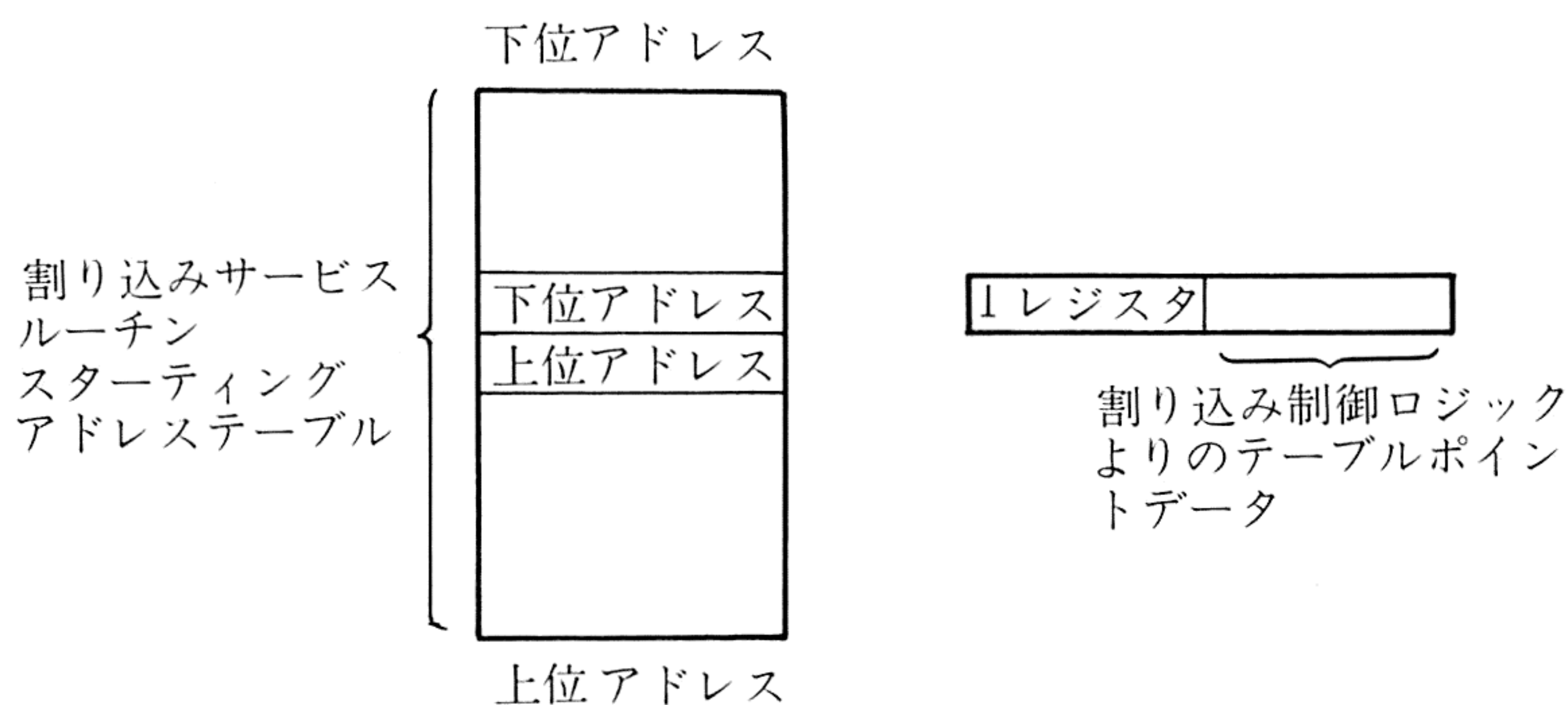



図14.1　割り込みアドレステーブル

テーブルの第一バイトは，アドレスの下位８ビットを格納します。 ユーザは，割り込みが受付けられる前に使用される割り込みチャネルに対応するサービスルーチンのスタートアドレスでこのテーブルをうめておく必要があります。

割り込みが発生し，割り込み制御ロジックよりポインタ下位８ビットが CPU に与えられると，CPU は自動的に PC （プログラムカウンタ）をスタックに退避させ，指定されたテーブルよりスタートアドレスを読み込み，その番地へジャンプします。

14.2 割り込みテーブル

MKII の割り込み制御ロジックには優先割り込みコントローラを使用しており，8レベルのプライオリティ付割り込みが使えるようになっています。

割り込みサービスルーチンスタートアドレステーブルは **N80-BASIC** により 8000H 番地～800FH 番地に自動的に配置されます。各割り込みチャンネルとそのテーブルのアドレスとの対応は表 14.1 のようになります。

表14．1　割り込みアドレス表

優先順位	スタートアドレステーブルアドレス	アドレステーブルの内容	チャネル機能
高	8000H 8001H	下位アドレス 上位 〃	RS-232C（本体内部で使用 シリアルチャネル）
↑	8002H 8003H	下位 〃 上位 〃	VRTC（本体内部で使用 垂直帰線信号）
	8004H 8005H	下位 〃 上位 〃	CLOCK（本体内部で使用 リアルタイムクロック）
	8006H 8007H	下位 〃 上位 〃	$\overline{\text{INT4}}$（スロットバスに開放）
	8008H 8009H	下位 〃 上位 〃	$\overline{\text{INT3}}$（スロットバスに開放）
	800AH 800BH	下位 〃 上位 〃	$\overline{\text{INT2}}$（スロットバスに開放）
	800CH 800DH	下位 〃 上位 〃	$\overline{\text{INT1}}$（スロットバスに開放 FDCINT1 FDC用リザーブ）
低	800EH 800FH	下位 〃 上位 〃	$\overline{\text{INT0}}$（スロットバスに開放 FDCINT2 FDC用リザーブ）

（注　意）**MKII** の割り込み優先順位は PC-8012 と一部ちがうので注意してください。

第15章
漢字の使い方

MKⅡ で漢字を扱うためには，漢字ROMボード（**PC-8001MKⅡ-01**）を**MKⅡ**の中に組み込まなくてはなりません。漢字ROMボードの取り付け方は，**PC-8001MKⅡ-01**に付いている説明書に詳しく書いてあります。

15.1 漢字をディスプレイに表示する

漢字は，グラフィック画面に表示されます。漢字を表示させるには，**CMD PUT** を用います。

(書　式) **CMD PUT (Hx, Hy), KANJI(〈漢字JISコード〉)〔,〈機能〉〕〔,〈フォアグラウンドカラーナンバ〉〕〔,〈バックグラウンドカラーナンバ〉〕**

(Sample) 画面に「愛」を表示させる。

```
CMD SCREEN 3
CONSOLE ,, 0, 1
CMD CLS 3
CMD PUT(152, 92), KANJI (&H3026), PSET, 2, 0
```

「愛」の漢字JISコード

この例では，320×200 ピクセルのカラーモード 2で，漢字JISコードが 3026H の「愛」を画面の中央に表示させています。漢字1文字は16ピクセル×16ピクセルで表わしますから，1画面にモノクロモード，アトリビュートカラーモードでは，30文字×12行，4色カラーモードでは20文字×12行まで表示することができます。

(参　照) **CMD PUT** について詳しくは N80-BASIC リファレンスマニ

ュアルをごらんください。

(Sample) 画面に漢字 JIS コード表を表示する。

```
10 CMD SCREEN 2
20 CMD COLOR 2
30 AD=&H3020
40 CMD CLS 3
50 LOCATE 0, 0 : PRINT "JIS CODE=&H";HEX$(AD)
60   FOR Y=22 TO 122 STEP 20
70     FOR X=2 TO 302 STEP 20
80       CMD PUT (X, Y), KANJI (AD)
90     AD=AD+1
100  NEXT
110 NEXT
120 AD=AD+160
130 IF AD> &H735F THEN END
140 GOTO 40
```

15.2 漢字をプリンタに出力する

PC-8822 は漢字 ROM ボードが実装されていますから，漢字コードを送ることによって漢字の印字ができます。(**PC-8821** にオプションの漢字 ROM ボード **PC-8021-01** を組み込んだ場合も同様です。)

PC-8822 に，ESC コードと "K" に続いて，漢字コードを2バイトにわけて送ります。

(Sample)
```
lprint chr$(27) ; "K" ; chr$(&H34) ; chr$(&h41); chr$(&h
3b) ; chr$(&h7a)
```

(注　意) LPRINT CHR$(x) を使用して送る場合 **N80-BASIC** のインタープリタにおいて，コード (&H9) によっては，他のコードを出力したり，ある文字数以上送ると自動的に改行されてしまい不都合が生じますので，次のサブルーチンによってデータを送ることをおすすめします。

```
100 ' PUT A CHARACTER
110 OUT &H10, C : OUT &H40, 0 : OUT &H40, 1
120 A=INP(&H40)AND 1 : IF A=0 THEN RETURN ELSE 120
```

上のサブルーチンを使用して次のような形式で呼びます。

C=x : GOSUB100

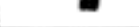
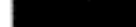
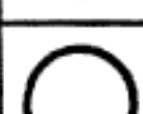
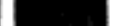
（xの値が，コードとして出力される。）

つまり

LPRINT CHR$(x) ;

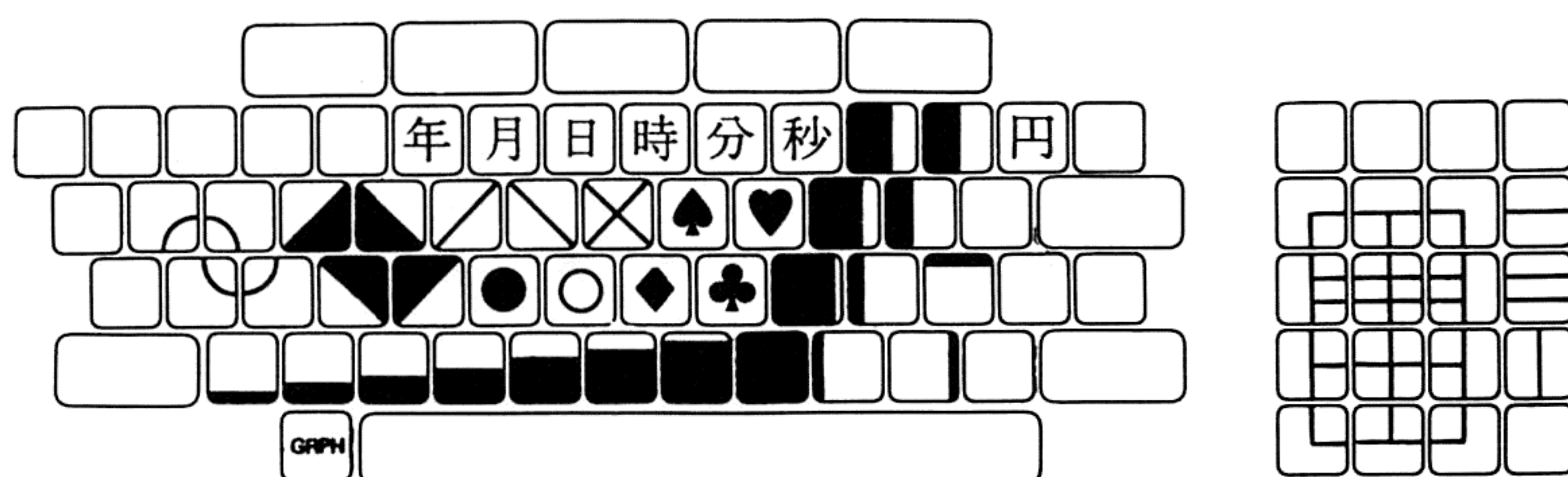
|||

C=x : GOSUB100

となっています。

PC-8023C に漢字を印字する場合は，グラフィック画面のコピーと同様に，漢字を画面に表示し　**CMD COPY 2** を実行します。

付　録

付録１　グラフィックシンボル

年 月 日 時 分 秒 円
GRPH

付録２　キャラクタコード

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		D_E		0	@	P		p	▁	┴		ー	タ	ミ	═	╳
1	S_H	D_1	!	1	A	Q	a	q	▂	┬	。	ア	チ	ム	╞	円
2	S_X	D_2	"	2	B	R	b	r	▃	┤	「	イ	ツ	メ	╪	年
3	E_X	D_3	#	3	C	S	c	s	▄	├	」	ウ	テ	モ	╡	月
4	E_T	D_4	$	4	D	T	d	t	▅	▔	、	エ	ト	ヤ	◢	日
5	E_Q	N_K	%	5	E	U	e	u	▆	━	・	オ	ナ	ユ	◣	時
6	A_K	S_N	&	6	F	V	f	v	▇	┃	ヲ	カ	ニ	ヨ	◥	分
7	B_L	E_B	'	7	G	W	g	w	█	▕	ァ	キ	ヌ	ラ	◤	秒
8	B_S	C_N	(	8	H	X	h	x	▏	┌	ィ	ク	ネ	リ	♠	
9	H_T	E_M	)	9	I	Y	i	y	▎	┐	ゥ	ケ	ノ	ル	♥	
A	L_F	S_B	*	:	J	Z	j	z	▍	└	ェ	コ	ハ	レ	♦	
B	H_M	E_C	+	;	K	[	k	{	▌	┘	ォ	サ	ヒ	ロ	♣	
C	C_L	→	,	<	L	¥	l	¦	▋	╭	ャ	シ	フ	ワ	●	
D	C_R	←	—	=	M	]	m	}	▊	╮	ュ	ス	ヘ	ン	○	
E	S_O	↑	.	>	N	^	n	~	▉	╰	ョ	セ	ホ	゛	╱	
F	S_I	↓	/	?	O	_	o		┼	╯	ッ	ソ	マ	゜	╲	

付録3　ジャンパスイッチと ディップスイッチの使い方

ディップスイッチ1　未使用

ディップスイッチ2（MEMORY WAIT スイッチ）
on……メモリリードサイクルに，1 WAIT動作をおこなう。（アクセスタイムが300nsec.より遅いROMを使用する場合）
off……通常

（注　意）PC-8012-03を使用する場合はディップスイッチの 2 をonにして下さい。

ディップスイッチ3～7
（ユーザは使用できません。）

ディップスイッチ8（ROM選択スイッチ）
on……N-BASICを選択
off……N_{80}-BASICを選択

ON ↑ 1 2 3 4 5 6 7 8　on / off

MODE SELECT

8 7 6 5 4 3 2 1
RS-232C

ボーレート
（bit per second）

	ボーレートファクタ16のとき	ボーレートファクタ64のとき
ジャンパスイッチ1	75	—
ジャンパスイッチ2	150	—
ジャンパスイッチ3	300	75
ジャンパスイッチ4	600	150
ジャンパスイッチ5	1200	300
ジャンパスイッチ6	2400	600
ジャンパスイッチ7	4800	1200
ジャンパスイッチ8	9600	2400

（注　意）ボーレート切り換え用のジャンパスイッチは，PC-8801 のものとは逆にならんでいるので注意してください。

付録4　エラーメッセージ

ERR コード	エラーメッセージ	意　味
1	NEXT without FOR	FOR－NEXTが正しく対応していない。(NEXTが多すぎる)
2	Syntax error	文法がまちがっている。プログラム中に規定のステートメント以外のものがある。
3	RETURN without GOSUB	GOSUB－RETURN が 正しく 対応していない。(RETURNだけがある。)
4	Out of Data	READ文で読まれるべきデータがDATA文の中に用意されていない。
5	Illegal function call	ステートメントの機能の呼び方がまちがっている。
6	Over flow	入力された数値や演算結果が許容される範囲を外れている。
7	Out of memory	メモリ容量が足りなくなった。(プログラムが長すぎる，配列が大きすぎる等)
8	Undefined line number	必要とされるプログラム行(GOTOの飛び先など)が定義されていない。
9	Subscript out of range	配列変数の添字が規定の範囲内にない。
10	Redimensioned array	同じ配列を再定義している。(同じDIM文が複数回実行された。)
11	Division by zero	0による割算が実行された。
12	Illegal direct	ダイレクト・ステートメントとして使えないコマンドが入力された。
13	Type mismatch	代入文などで式の左右の型が一致していない。(数値とストリングなど)
14	Out of string space	CLEAR文などで指定した，ストリング変数用メモリエリアが足りなくなった。
15	String too long	ストリング(引用符で囲まれた文字列)が長すぎる。(255文字をこえた)
16	String formura too complex	文字式が複雑すぎる。(カッコのネスティングレベルが多すぎる等)
17	Can't continue	CONTコマンドを入力しても続行できない。(ポインタが破壊されている。)
18	Undefined usr function	参照されたユーザー関数の定義(DEF文による)がなされていない。
19	No RESUME	エラー処理後，プログラムの実行を再開することができない。
20	RESUME without error	エラーとRESUMEが対応していない。(エラーがないのにRESUMEした。)
21 (未定義)	Unprintable error	メッセージの定義されていないエラー。

ERR コード	エラーメッセージ	意　　味
22	Missing operand	ステートメント中に必要とされるパラメータなどが指定されていない。
23	Line buffer overflow	BASICが，行を入力する際，一行の長さが有効範囲を越している。
24	Position not on Screen	指定したカーソル位置などが，画面の範囲外になっている
25	Bad File Data	ファイル上にあるデータの形式がまちがっている。
26	Disk BASIC Feature	ディスクが接続されていないとき，DISK BASICモードの命令を実行した。
27	Communications Buffer overflow	周辺機器との入出力のためのバッファがオーバーフローした。
28	Port not initialized	インタフェース用のLSIの機能設定がなされていない。
29	Tape read ERROR	カセットからの入力が正しく行なわれていない。(テープの読み誤りなど)

DISK-BASIC モードで追加されるエラーメッセージ

ERR コード	エラーメッセージ	意　　味
50	FIELD overflow	FIELD文で256バイト以上の文字を割り当てた。
51	Internal error	BASIC内部でのエラー
52	Bad file number	オープンされていないファイル・ナンバーをアクセスした。
53	File not found	LOAD, KILL, OPENなどで，存在しないファイルをアクセスした。
54	Bad file mode	シーケンシャル・ファイルでOPENしたファイルに対して，ランダム・アクセスをした。またはその逆。
55	File already open	すでに開かれているファイルに対してOPENやKILLを実行した。
56	Disk not mounted	mountされていないディスクに対してアクセスした。
57	Disk I/O error	ディスクにリード　ライトエラーが生じ読み直しても修正できなくなった。
58	File already exists	NAMEによって定義されたファイルネームが，すでに登録されている。
59	Disk already mounted	MOUNTされているディスクに対して MOUNTを行なった。

ERR コード	エラーメッセージ	意　　味
60	Disk Full	ディスク上のすべての場所を使い切った。
61	Input past end	ファイルのすべてのデータを読み尽した後にINPUTが実行された。
62	Bad file name	不適当なファイルネームを使った。
63	Direct statement in file	アスキー・フォーマットでのLOAD中にダイレクト・ステートメントがあった。
64	Bad allocation table	FATがこわれた。
65	Bad drive number	不適当なドライブネームを使った。
66	Bad surface/track/sector	サーフェス，トラック，セクタ番号の指定が誤っている。(DSKO$, DSKI$)
67	Deleted record	消去済のレコードをアクセスしようとした。
68	Rename across disks	ファイル名の付け換えで違ったドライブナンバーを指定した。
69	Sequential after PUT	PUT文実行後，シーケンシャルファイルをアクセスしようとした。
70	Sequential I/O Only	シーケンシャル入出力以外は行ってはならない。
71	File not open	まだ開かれていないファイルに対してアクセスした。
72	File write protected	ライトプロテクトがかかっているファイルに対して書き込んだ。
73	Disk offline	入出力の可能な状態でないフロッピィディスクをアクセスしようとした。

付録5　演算順位

優先順位	演　　算
1	カッコで囲まれたもの
2	関数
3	指数(べき乗)^
4	負号(－)
5	*，/
6	¥
7	MOD
8	+，－
9	関係演算子(<，>，＝など)
10	NOT
11	AND
12	OR
13	XOR
14	IMP
15	EQV

付録6　コントロールコード

キー入力	動　　　　作
CTRL + B	カーソルを1項目ごと左へ移動させます。
CTRL + E	カーソルの位置から，その行の終りまでをまっ消します。
CTRL + G	ブザーを鳴らします。
CTRL + H	1文字を削除します。 INS DEL と同じ働きです。
CTRL + I	タブ位置までカーソルを移動させます。タブ位置は8桁ごとに設定されています。 TAB キーと同じ働きです。
CTRL + J	ラインフィード(LFコード)を挿入します。
CTRL + K	カーソルをホームポジション（左上スミ）に 移動させます。 SHIFT + HOME CLR と同じです。
CTRL + L	テキスト画面をクリアします。 HOME CLR と同じです。
CTRL + M	RETURN と同じです。
CTRL + N	カーソルを1項目ごとに右へ移動させます。
CTRL + R	スペースを挿入します。 SHIFT + INS DEL と同じです。
ESC	実行を一時中断します。
⇆	カーソルを1つ右へ移動させます。
SHIFT + ⇆	カーソルを1つ左へ移動させます。
⇵	カーソルを上の行へ移動させます。
SHIFT + ⇵	カーソルを下の行へ移動させます。

付録7　漢字コード

半角文字一覧表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0020		!	"	#	$	%	&	'	(	)	*	+	,	－	.	/
0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	—
0060		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0070	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	
0080		。	「	」	、	・	を	ぁ	ぃ	ぅ	ぇ	ぉ	ゃ	ゅ	ょ	っ
0090	ー	あ	い	う	え	お	か	き	く	け	こ	さ	し	す	せ	そ
00A0		。	「	」		・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
00B0	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
00C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
00D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	゜
00E0	た	ち	つ	て	と	な	に	ぬ	ね	の	は	ひ	ふ	へ	ほ	ま
00F0	み	む	め	も	や	ゆ	よ	ら	り	る	れ	ろ	わ	ん	゛	゜
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

1/4 角文字一覧表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0100		S_H	S_X	E_X	E_T	E_Q	A_K	B_L	B_S	H_T	L_F	H_M	C_L	C_R	S_O	S_I
0110	D_E	D_1	D_2	D_3	D_4	N_K	S_N	E_B	C_N	E_M	S_B	E_C	→	←	↑	↓
0120		!	″	#	$	%	&	'	(	)	*	+	,	−	.	/
0130	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0140	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0150	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	∧	—
0160		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0170	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	
0180																+
0190													╭	╮	╰	╯
01A0		。	「	」	、	・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
01B0	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
01C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
01D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	゜
01E0	═	╞	╪	╡					♠	♥	♦	♣	●	○	╱	╲
01F0	×	円	年	月	日	時	分	秒								
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
記号	2120			、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨
	2130	＾	￣	＿	ヽ	ヾ	ゝ	ゞ	〃	仝	々	〆	〇	ー	―	‐	／
	2140	＼	～	‖	｜	…	‥	‘	’	“	”	（	）	〔	〕	［	］
	2150	｛	｝	〈	〉	《	》	「	」	『	』	【	】	＋	－	±	×
	2160	÷	＝	≠	＜	＞	≦	≧	∞	∴	♂	♀	°	′	″	℃	￥
	2170	＄	¢	£	％	＃	＆	＊	＠	§	☆	★	○	●	◎	◇	
	2220		◆	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	＝	
英・数字	2330	0	1	2	3	4	5	6	7	8	9						
	2340		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	2350	P	Q	R	S	T	U	V	W	X	Y	Z					
	2360		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	2370	p	q	r	s	t	u	v	w	x	y	z					
ひらがな	2420		ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く
	2430	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
	2440	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
	2450	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
	2460	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
	2470	ゐ	ゑ	を	ん												
カタカナ	2520		ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク
	2530	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ
	2540	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ
	2550	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ
	2560	ム	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ
	2570	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ									
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

※コードは16進で表現されています。
例えば〝B″のコードは2340H＋2＝2342Hとなります。

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ギリシア文字	2620		Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
	2630	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω							
	2640		α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
	2650	π	ρ	σ	τ	υ	φ	χ	ψ	ω							
ロシア文字	2720		А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
	2730	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
	2740	Ю	Я														
	2750		а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н
	2760	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
	2770	ю	я														
ア	3020		亜	唖	娃	阿	哀	愛	挨	姶	逢	葵	茜	穐	悪	握	渥
	3030	旭	葦	芦	鯵	梓	圧	斡	扱	宛	姐	虻	飴	絢	綾	鮎	或
	3040	粟	袷	安	庵	按	暗	案	闇	鞍	杏						
イ	3040											以	伊	位	依	偉	囲
	3050	夷	委	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃
	3060	萎	衣	謂	違	遺	医	井	亥	域	育	郁	磯	一	壱	溢	逸
	3070	稲	茨	芋	鰯	允	印	咽	員	因	姻	引	飲	淫	胤	蔭	
	3120		院	陰	隠	韻	吋										
ウ	3120							右	宇	烏	羽	迂	雨	卯	鵜	窺	丑
	3130	碓	臼	渦	嘘	唄	欝	蔚	鰻	姥	厩	浦	瓜	閏	噂	云	運
	3140	雲															
エ	3140		荏	餌	叡	営	嬰	影	映	曳	栄	永	泳	洩	瑛	盈	穎
	3150	頴	英	衛	詠	鋭	液	疫	益	駅	悦	謁	越	閲	榎	厭	円
	3160	園	堰	奄	宴	延	怨	掩	援	沿	演	炎	焔	煙	燕	猿	縁
	3170	艶	苑	薗	遠	鉛	鴛	塩									
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
オ	3170								於	汚	甥	凹	央	奥	往	応	
	3220		押	旺	横	欧	殴	王	翁	襖	鴬	鴎	黄	岡	沖	荻	億
	3230	屋	憶	臆	桶	牡	乙	俺	卸	恩	温	穏	音				
カ	3230													下	化	仮	何
	3240	伽	価	佳	加	可	嘉	夏	嫁	家	寡	科	暇	果	架	歌	河
	3250	火	珂	禍	禾	稼	箇	花	苛	茄	荷	華	菓	蝦	課	嘩	貨
	3260	迦	過	霞	蚊	俄	峨	我	牙	画	臥	芽	蛾	賀	雅	餓	駕
	3270	介	会	解	回	塊	壊	廻	快	怪	悔	恢	懐	戒	拐	改	
	3320		魁	晦	械	海	灰	界	皆	絵	芥	蟹	開	階	貝	凱	劾
	3330	外	咳	害	崖	慨	概	涯	碍	蓋	街	該	鎧	骸	浬	馨	蛙
	3340	垣	柿	蠣	鈎	劃	嚇	各	廓	拡	撹	格	核	殻	獲	確	穫
	3350	覚	角	赫	較	郭	閣	隔	革	学	岳	楽	額	顎	掛	笠	樫
	3360	橿	梶	鰍	潟	割	喝	恰	括	活	渇	滑	葛	褐	轄	且	鰹
	3370	叶	椛	樺	鞄	株	兜	竈	蒲	釜	鎌	噛	鴨	栢	茅	萱	
	3420		粥	刈	苅	瓦	乾	侃	冠	寒	刊	勘	勧	巻	喚	堪	姦
	3430	完	官	寛	干	幹	患	感	慣	憾	換	敢	柑	桓	棺	款	歓
	3440	汗	漢	澗	灌	環	甘	監	看	竿	管	簡	緩	缶	翰	肝	艦
	3450	莞	観	諫	貫	還	鑑	間	閑	関	陥	韓	館	舘	丸	含	岸
	3460	巌	玩	癌	眼	岩	翫	贋	雁	頑	顔	願					
キ	3460												企	伎	危	喜	器
	3470	基	奇	嬉	寄	岐	希	幾	忌	揮	机	旗	既	期	棋	棄	
	3520		機	帰	毅	気	汽	畿	祈	季	稀	紀	徽	規	記	貴	起
	3530	軌	輝	飢	騎	鬼	亀	偽	儀	妓	宜	戯	技	擬	欺	犠	疑
	3540	祇	義	蟻	誼	議	掬	菊	鞠	吉	吃	喫	桔	橘	詰	砧	杵
	3550	黍	却	客	脚	虐	逆	丘	久	仇	休	及	吸	宮	弓	急	救
	3560	朽	求	汲	泣	灸	球	究	窮	笈	級	糾	給	旧	牛	去	居
	3570	巨	拒	拠	挙	渠	虚	許	距	鋸	漁	禦	魚	亨	享	京	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	3620		供	侠	僑	兇	競	共	凶	協	匡	卿	叫	喬	境	峡	強
	3630	彊	怯	恐	恭	挾	教	橋	況	狂	狭	矯	胸	脅	興	蕎	郷
	3640	鏡	響	饗	驚	仰	凝	堯	暁	業	局	曲	極	玉	桐	粁	僅
	3650	勤	均	巾	錦	斤	欣	欽	琴	禁	禽	筋	緊	芹	菌	衿	襟
	3660	謹	近	金	吟	銀											
ク	3660						九	倶	句	区	狗	玖	矩	苦	軀	駆	駈
	3670	駒	具	愚	虞	喰	空	偶	寓	遇	隅	串	櫛	釧	屑	屈	
	3720		掘	窟	沓	靴	轡	窪	熊	隈	粂	栗	繰	桑	鍬	勲	君
	3730	薫	訓	群	軍	郡											
ケ	3730						卦	袈	祁	係	傾	刑	兄	啓	圭	珪	型
	3740	契	形	径	恵	慶	慧	憩	掲	携	敬	景	桂	渓	畦	稽	系
	3750	経	継	繋	罫	茎	荊	蛍	計	詣	警	軽	頚	鶏	芸	迎	鯨
	3760	劇	戟	撃	激	隙	桁	傑	欠	決	潔	穴	結	血	訣	月	件
	3770	倹	倦	健	兼	券	剣	喧	圏	堅	嫌	建	憲	懸	拳	捲	
	3820		検	権	牽	犬	献	研	硯	絹	県	肩	見	謙	賢	軒	遣
	3830	鍵	険	顕	験	鹸	元	原	厳	幻	弦	減	源	玄	現	絃	舷
	3840	言	諺	限													
コ	3840				乎	個	古	呼	固	姑	孤	己	庫	弧	戸	故	枯
	3850	湖	狐	糊	袴	股	胡	菰	虎	誇	跨	鈷	雇	顧	鼓	五	互
	3860	伍	午	呉	吾	娯	後	御	悟	梧	檎	瑚	碁	語	誤	護	醐
	3870	乞	鯉	交	佼	侯	候	倖	光	公	功	効	勾	厚	口	向	
	3920		后	喉	坑	垢	好	孔	孝	宏	工	巧	巷	幸	広	庚	康
	3930	弘	恒	慌	抗	拘	控	攻	昂	晃	更	杭	校	梗	構	江	洪
	3940	浩	港	溝	甲	皇	硬	稿	糠	紅	紘	絞	綱	耕	考	肯	肱
	3950	腔	膏	航	荒	行	衡	講	貢	購	郊	酵	鉱	礦	鋼	閤	降
	3960	項	香	高	鴻	剛	劫	号	合	壕	拷	濠	豪	轟	麹	克	刻
	3970	告	国	穀	酷	鵠	黒	獄	漉	腰	甑	忽	惚	骨	狛	込	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	3A20		此	頃	今	困	坤	墾	婚	恨	懇	昏	昆	根	梱	混	痕
	3A30	紺	艮	魂													
サ	3A30				些	佐	叉	唆	嵯	左	差	査	沙	瑳	砂	詐	鎖
	3A40	裟	坐	座	挫	債	催	再	最	哉	塞	妻	宰	彩	才	採	栽
	3A50	歳	済	災	采	犀	砕	砦	祭	斎	細	菜	裁	載	際	剤	在
	3A60	材	罪	財	冴	坂	阪	堺	榊	肴	咲	崎	埼	碕	鷺	作	削
	3A70	咋	搾	昨	朔	柵	窄	策	索	錯	桜	鮭	笹	匙	冊	刷	
	3B20		察	拶	撮	擦	札	殺	薩	雑	皐	鯖	捌	錆	鮫	皿	晒
	3B30	三	傘	参	山	惨	撒	散	桟	燦	珊	産	算	纂	蚕	讃	賛
	3B40	酸	餐	斬	暫	残											
シ	3B40						仕	仔	伺	使	刺	司	史	嗣	四	士	始
	3B50	姉	姿	子	屍	市	師	志	思	指	支	孜	斯	施	旨	枝	止
	3B60	死	氏	獅	祉	私	糸	紙	紫	肢	脂	至	視	詞	詩	試	誌
	3B70	諮	資	賜	雌	飼	歯	事	似	侍	児	字	寺	慈	持	時	
	3C20		次	滋	治	爾	璽	痔	磁	示	而	耳	自	蒔	辞	汐	鹿
	3C30	式	識	鴫	竺	軸	宍	雫	七	叱	執	失	嫉	室	悉	湿	漆
	3C40	疾	質	実	蔀	篠	偲	柴	芝	屢	蕊	縞	舎	写	射	捨	赦
	3C50	斜	煮	社	紗	者	謝	車	遮	蛇	邪	借	勺	尺	杓	灼	爵
	3C60	酌	釈	錫	若	寂	弱	惹	主	取	守	手	朱	殊	狩	珠	種
	3C70	腫	趣	酒	首	儒	受	呪	寿	授	樹	綬	需	囚	収	周	
	3D20		宗	就	州	修	愁	拾	洲	秀	秋	終	繍	習	臭	舟	蒐
	3D30	衆	襲	讐	蹴	輯	週	酋	酬	集	醜	什	住	充	十	従	戎
	3D40	柔	汁	渋	獣	縦	重	銃	叔	夙	宿	淑	祝	縮	粛	塾	熟
	3D50	出	術	述	俊	峻	春	瞬	竣	舜	駿	准	循	旬	楯	殉	淳
	3D60	準	潤	盾	純	巡	遵	醇	順	処	初	所	暑	曙	渚	庶	緒
	3D70	署	書	薯	藷	諸	助	叙	女	序	徐	恕	鋤	除	傷	償	
	3E20		勝	匠	升	召	哨	商	唱	嘗	奨	妾	娼	宵	将	小	少
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	3E30	尚	庄	床	廠	彰	承	抄	招	掌	捷	昇	昌	昭	晶	松	梢
	3E40	樟	樵	沼	消	渉	湘	焼	焦	照	症	省	硝	礁	祥	称	章
	3E50	笑	粧	紹	肖	菖	蒋	蕉	衝	裳	訟	証	詔	詳	象	賞	醤
	3E60	鉦	鍾	鐘	障	鞘	上	丈	丞	乗	冗	剰	城	場	壌	嬢	常
	3E70	情	擾	条	杖	浄	状	畳	穣	蒸	譲	醸	錠	嘱	埴	飾	
	3F20		拭	植	殖	燭	織	職	色	触	食	蝕	辱	尻	伸	信	侵
	3F30	唇	娠	寝	審	心	慎	振	新	晋	森	榛	浸	深	申	疹	真
	3F40	神	秦	紳	臣	芯	薪	親	診	身	辛	進	針	震	人	仁	刃
	3F50	塵	壬	尋	甚	尽	腎	訊	迅	陣	靭						
ス	3F50											笥	諏	須	酢	図	厨
	3F60	逗	吹	垂	帥	推	水	炊	睡	粋	翠	衰	遂	酔	錐	錘	随
	3F70	瑞	髄	崇	嵩	数	枢	趨	雛	据	杉	椙	菅	頗	雀	裾	
	4020		澄	摺	寸												
セ	4020					世	瀬	畝	是	凄	制	勢	姓	征	性	成	政
	4030	整	星	晴	棲	栖	正	清	牲	生	盛	精	聖	声	製	西	誠
	4040	誓	請	逝	醒	青	静	斉	税	脆	隻	席	惜	戚	斥	昔	析
	4050	石	積	籍	績	脊	責	赤	跡	蹟	碩	切	拙	接	摂	折	設
	4060	窃	節	説	雪	絶	舌	蝉	仙	先	千	占	宣	専	尖	川	戦
	4070	扇	撰	栓	栴	泉	浅	洗	染	潜	煎	煽	旋	穿	箭	線	
	4120		繊	羨	腺	舛	船	薦	詮	賎	践	選	遷	銭	銑	閃	鮮
	4130	前	善	漸	然	全	禅	繕	膳	糎							
ソ	4130										噌	塑	岨	措	曾	曽	楚
	4140	狙	疏	疎	礎	祖	租	粗	素	組	蘇	訴	阻	遡	鼠	僧	創
	4150	双	叢	倉	喪	壮	奏	爽	宋	層	匝	惣	想	捜	掃	挿	掻
	4160	操	早	曹	巣	槍	槽	漕	燥	争	痩	相	窓	糟	総	綜	聡
	4170	草	荘	葬	蒼	藻	装	走	送	遭	鎗	霜	騒	像	増	憎	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4220		臓	蔵	贈	造	促	側	則	即	息	捉	束	測	足	速	俗
	4230	属	賊	族	続	卒	袖	其	揃	存	孫	尊	損	村	遜		
タ	4230															他	多
	4240	太	汰	詑	唾	堕	妥	惰	打	柁	舵	楕	陀	駄	騨	体	堆
	4250	対	耐	岱	帯	待	怠	態	戴	替	泰	滞	胎	腿	苔	袋	貸
	4260	退	逮	隊	黛	鯛	代	台	大	第	醍	題	鷹	滝	瀧	卓	啄
	4270	宅	托	択	拓	沢	濯	琢	託	鐸	濁	諾	茸	凧	蛸	只	
	4320		叩	但	達	辰	奪	脱	巽	竪	辿	棚	谷	狸	鱈	樽	誰
	4330	丹	単	嘆	坦	担	探	旦	歎	淡	湛	炭	短	端	箪	綻	耽
	4340	胆	蛋	誕	鍛	団	壇	弾	断	暖	檀	段	男	談			
チ	4340														値	知	地
	4350	弛	恥	智	池	痴	稚	置	致	蜘	遅	馳	築	畜	竹	筑	蓄
	4360	逐	秩	窒	茶	嫡	着	中	仲	宙	忠	抽	昼	柱	注	虫	衷
	4370	註	酎	鋳	駐	樗	瀦	猪	苧	著	貯	丁	兆	凋	喋	寵	
	4420		帖	帳	庁	弔	張	彫	徴	懲	挑	暢	朝	潮	牒	町	眺
	4430	聴	脹	腸	蝶	調	諜	超	跳	銚	長	頂	鳥	勅	捗	直	朕
	4440	沈	珍	賃	鎮	陳											
ツ	4440						津	墜	椎	槌	追	鎚	痛	通	塚	栂	掴
	4450	槻	佃	漬	柘	辻	蔦	綴	鍔	椿	潰	坪	壺	嬬	紬	爪	吊
	4460	釣	鶴														
テ	4460			亭	低	停	偵	剃	貞	呈	堤	定	帝	底	庭	廷	弟
	4470	悌	抵	挺	提	梯	汀	碇	禎	程	締	艇	訂	諦	蹄	逓	
	4520		邸	鄭	釘	鼎	泥	摘	擢	敵	滴	的	笛	適	鏑	溺	哲
	4530	徹	撤	轍	迭	鉄	典	塡	天	展	店	添	纏	甜	貼	転	顛
	4540	点	伝	殿	澱	田	電										
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ト	4540							兎	吐	堵	塗	妬	屠	徒	斗	杜	渡
	4550	登	菟	賭	途	都	鍍	砥	礪	努	度	土	奴	怒	倒	党	冬
	4560	凍	刀	唐	塔	塘	套	宕	島	嶋	悼	投	搭	東	桃	檮	棟
	4570	盗	淘	湯	濤	灯	燈	当	痘	禱	等	答	筒	糖	統	到	
	4620		董	蕩	藤	討	謄	豆	踏	逃	透	鐙	陶	頭	騰	闘	働
	4630	動	同	堂	導	憧	撞	洞	瞳	童	胴	萄	道	銅	峠	鴇	匿
	4640	得	徳	瀆	特	督	禿	篤	毒	独	読	栃	橡	凸	突	椴	届
	4650	鳶	苫	寅	酉	瀞	噸	屯	惇	敦	沌	豚	遁	頓	呑	曇	鈍
ナ	4660	奈	那	内	乍	凪	薙	謎	灘	捺	鍋	楢	馴	縄	畷	南	楠
	4670	軟	難	汝													
ニ	4670				二	尼	弐	邇	匂	賑	肉	虹	廿	日	乳	入	
	4720		如	尿	韮	任	妊	忍	認								
ヌ	4720									濡							
ネ	4720										禰	祢	寧	葱	猫	熱	年
	4730	念	捻	撚	燃	粘											
ノ	4730						乃	廼	之	埜	嚢	悩	濃	納	能	脳	膿
	4740	農	覗	蚤													
ハ	4740				巴	把	播	覇	杷	波	派	琶	破	婆	罵	芭	馬
	4750	俳	廃	拝	排	敗	杯	盃	牌	背	肺	輩	配	倍	培	媒	梅
	4760	楳	煤	狽	買	売	賠	陪	這	蝿	秤	矧	萩	伯	剥	博	拍
	4770	柏	泊	白	箔	粕	舶	薄	迫	曝	漠	爆	縛	莫	駁	麦	
	4820		函	箱	硲	箸	肇	筈	櫨	幡	肌	畑	畠	八	鉢	溌	発
	4830	醗	髪	伐	罰	抜	筏	閥	鳩	噺	塙	蛤	隼	伴	判	半	反
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4840	叛	帆	搬	斑	板	氾	汎	版	犯	班	畔	繁	般	藩	販	範
	4850	釆	煩	頒	飯	挽	晩	番	盤	磐	蕃	蛮					
ヒ	4850												匪	卑	否	妃	庇
	4860	彼	悲	扉	批	披	斐	比	泌	疲	皮	碑	秘	緋	罷	肥	被
	4870	誹	費	避	非	飛	樋	簸	備	尾	微	枇	毘	琵	眉	美	
	4920		鼻	柊	稗	匹	疋	髭	彦	膝	菱	肘	弼	必	畢	筆	逼
	4930	桧	姫	媛	紐	百	謬	俵	彪	標	氷	漂	瓢	票	表	評	豹
	4940	廟	描	病	秒	苗	錨	鋲	蒜	蛭	鰭	品	彬	斌	浜	瀕	貧
	4950	賓	頻	敏	瓶												
フ	4950					不	付	埠	夫	婦	富	冨	布	府	怖	扶	敷
	4960	斧	普	浮	父	符	腐	膚	芙	譜	負	賦	赴	阜	附	侮	撫
	4970	武	舞	葡	蕪	部	封	楓	風	葺	蕗	伏	副	復	幅	服	
	4A20		福	腹	複	覆	淵	弗	払	沸	仏	物	鮒	分	吻	噴	墳
	4A30	憤	扮	焚	奮	粉	糞	紛	雰	文	聞						
ヘ	4A30											丙	併	兵	塀	幣	平
	4A40	弊	柄	並	蔽	閉	陛	米	頁	僻	壁	癖	碧	別	瞥	蔑	箆
	4A50	偏	変	片	篇	編	辺	返	遍	便	勉	娩	弁	鞭			
ホ	4A50														保	舗	鋪
	4A60	圃	捕	歩	甫	補	輔	穂	募	墓	慕	戊	暮	母	簿	菩	倣
	4A70	俸	包	呆	報	奉	宝	峰	峯	崩	庖	抱	捧	放	方	朋	
	4B20		法	泡	烹	砲	縫	胞	芳	萌	蓬	蜂	褒	訪	豊	邦	鋒
	4B30	飽	鳳	鵬	乏	亡	傍	剖	坊	妨	帽	忘	忙	房	暴	望	某
	4B40	棒	冒	紡	肪	膨	謀	貌	貿	鉾	防	吠	頬	北	僕	卜	墨
	4B50	撲	朴	牧	睦	穆	釦	勃	没	殆	堀	幌	奔	本	翻	凡	盆
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
マ	4B60	摩	磨	魔	麻	埋	妹	昧	枚	毎	哩	槙	幕	膜	枕	鮪	柾
	4B70	鱒	桝	亦	俣	又	抹	末	沫	迄	儘	繭	麿	万	慢	満	
	4C20		漫	蔓													
ミ	4C20				味	未	魅	巳	箕	岬	密	蜜	湊	蓑	稔	脈	妙
	4C30	粍	民	眠													
ム	4C30				務	夢	無	牟	矛	霧	鵡	椋	婿	娘			
メ	4C30														冥	名	命
	4C40	明	盟	迷	銘	鳴	姪	牝	滅	免	棉	綿	緬	面	麺		
モ	4C40															摸	模
	4C50	茂	妄	孟	毛	猛	盲	網	耗	蒙	儲	木	黙	目	杢	勿	餅
	4C60	尤	戻	籾	貰	問	悶	紋	門	匁							
ヤ	4C60										也	冶	夜	爺	耶	野	弥
	4C70	矢	厄	役	約	薬	訳	躍	靖	柳	藪	鑓					
ユ	4C70												愉	愈	油	癒	
	4D20		諭	輸	唯	佑	優	勇	友	宥	幽	悠	憂	揖	有	柚	湧
	4D30	涌	猶	猷	由	祐	裕	誘	遊	邑	郵	雄	融	夕			
ヨ	4D30														予	余	与
	4D40	誉	輿	預	傭	幼	妖	容	庸	揚	揺	擁	曜	楊	様	洋	溶
	4D50	熔	用	窯	羊	耀	葉	蓉	要	謡	踊	遙	陽	養	慾	抑	欲
	4D60	沃	浴	翌	翼	淀											
ラ	4D60						羅	螺	裸	来	莱	頼	雷	洛	絡	落	酪
	4D70	乱	卵	嵐	欄	濫	藍	蘭	覧								
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
リ	4D70									利	吏	履	李	梨	理	璃	
	4E20		痢	裏	裡	里	離	陸	律	率	立	葎	掠	略	劉	流	溜
	4E30	琉	留	硫	粒	隆	竜	龍	侶	慮	旅	虜	了	亮	僚	両	凌
	4E40	寮	料	梁	涼	猟	療	瞭	稜	糧	良	諒	遼	量	陵	領	力
	4E50	緑	倫	厘	林	淋	燐	琳	臨	輪	隣	鱗	麟				
ル	4E50													瑠	塁	涙	累
	4E60	類															
レ	4E60		令	伶	例	冷	励	嶺	怜	玲	礼	苓	鈴	隷	零	霊	麗
	4E70	齢	暦	歴	列	劣	烈	裂	廉	恋	憐	漣	煉	簾	練	聯	
	4F20		蓮	連	錬												
ロ	4F20					呂	魯	櫓	炉	賂	路	露	労	婁	廊	弄	朗
	4F30	楼	榔	浪	漏	牢	狼	籠	老	聾	蝋	郎	六	麓	禄	肋	録
	4F40	論															
ワ	4F40		倭	和	話	歪	賄	脇	惑	枠	鷲	亙	亘	鰐	詫	藁	蕨
	4F50	椀	湾	碗	腕												
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

付録8　予約語

ここにあげた予約語は，**N$_{80}$-BASIC, N-BASIC** で，特定の意味を持ち，変数名などに使うことはできません。

A　AND
ABS
ATN
ASC
AUTO
ATTR$

B　BCD$
BEEP

C　CONSOLE
CLOSE
CONT
CLEAR
CLOAD
CSAVE
CSRLIN
CINT
CSNG
CDBL
CVI
CVS
CVD
COS
CHR$
CMD
COLOR

D　DATA
DIM
DEFSTR
DEFINT
DEFSNG
DEFDBL
DSKO$
DEF
DELETE
DSKI$
DSKF
DEC
DATE$

E　END
ELSE
ERASE
ERROR
ERL
ERR
EXP
EOF
EQV

F　FORMAT
FOR
FIELD
FILES
FN
FRE
FIX
FPOS

G　GOTO
GO TO
GOSUB
GET

H　HEX$

I　INPUT
IF
INSTR
INT
INP
IMP
INIT
INKEY$
ISET
IRESET
IEEE

K　KILL
KEY

L　LET
LOCATE
LINE
LOAD
LSET
LPRINT
LLIST
LPOS
LISTEN
LIST
LFILES
LOG
LOC
LEN
LEFT$
LOF

M　MOUNT
MERGE
MOD

MKI$
MKS$
MKD$
MID$
MOTOR
MON
MAT

N NEXT
NAME
NEW
NOT

O OUT
ON
OPEN
OR
OCT$

P PUT
POKE
PRINT
POS
PEEK
PORT
POLL
PSET
PRESET
POINT

R READ
RUN
RESTONE
RETURN
REMOVE
REM
RESUME
RSET
RIGHT$
RND
RENUM
RBYTE

S STOP
SWAP
SET
SAVE
SPC(
STEP
SGN
SQR
SIN
STR$
STRING$
SPACE$
STATUS
SRQ

T TRON
TROFF
TAB(
TO
THEN
TAN
TERM
TALK
TIME$

U USING
USR

V VAL
VARPTR

W WIDTH
WAIT
WBYTE

X XOR

+
−
*
/
^
¥
"
>
=
<

付録9　I/Oマップ

I/Oアドレス0H～7FHまでと，下図のアミの部分は**MKII**が使用していますから，ユーザが使用できるアドレスは，下図のアミのない部分だけです。

アドレス	内容
80H～	（ユーザ使用可能）
C0H～	RS-232Cインタフェース I/Oアドレス
C8H～	（ユーザ使用可能）
D0H～	GP-IBインタフェース I/Oアドレス
E0H～	（アミ）システムコントロール I/Oアドレス
E8H～	（ユーザ使用可能）
F0H～FFH	（アミ）フロッピィディスク制御 I/Oアドレス

□ ユーザ使用可能 I/Oアドレスエリア

▨ **MKII**が使用しているI/Oアドレス

付録10　メモリマップ

FFFFH
RAM
64Kバイト
0H

7FFFH
ROM 2
16Kバイト
4000H
3FFFH
ROM 1
16Kバイト
0H

7FFFH
6000H
ROM 3
8Kバイト

BFFFH
G-V RAM
16Kバイト
8000H

オプション
RAM
32Kバイト

ROM 1, ROM 2……N_{80} -BASIC
ROM 3……増設用ROM

FFFFH
RAM
8000H
0H
ROM
G-V RAM

N_{80} -BASICで使われるメモリ

FFFFH
RAM
8000H
0H
ROM

N-BASICで使われるメモリ

付録11　μPD780ニーモニック↔機械語対照表

8ビット・ロード

X	I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX +d)	(IY +d)	(nn)	n
LD A, X	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n n	3E n
LD B, X			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d		06 n
LD C, X			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		0E n
LD D, X			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d		16 n
LD E, X			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
LD H, X			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d		26 n
LD L, X			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
LD (HL), X			77	70	71	72	73	74	75							36 n
LD (BC), X			02													
LD (DE), X			12													
LD (IX+d), X			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d n
LD (IY+d), X			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d n
LD (nn), X			32 n n													
LD I, X			ED 47													
LD R, X			ED 4F													

16ビット・ロード

X	AF	BC	DE	HL	SP	IX	IY	nn	(nn)
LD AF, X									
LD BC, X								01 n n	ED 4B n n
LD DE, X								11 n n	ED 5B n n
LD HL, X								21 n n	2A n n
LD SP, X				F9		DD F9	FD F9	31 n n	ED 7B n n
LD IX, X								DD 21 n n	DD 2A n n
LD IY, X								FD 21 n n	FD 2A n n
LD (nn), X		ED 43 n n	ED 53 n n	22 n n	ED 73 n n	DD 22 n n	FD 22 n n		
PUSH X	F5	C5	D5	E5		DD E5	FD E5		
POP X	F1	C1	D1	E1		DD E1	FD E1		

ブロック転送

LDI	ED A0
LDIR	ED B0
LDD	ED A8
LDDR	ED B8

ブロック・サーチ

CPI	ED A1
DPIR	ED B1
CPD	ED A9
CPDR	ED B9

8ビット算術論理演算

X	A	B	C	D	E	H	L	(HL)	(IX +d)	(IY +d)	n
ADD A, X	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
ADC A, X	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
SUB X	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
SBC A, X	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
AND X	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
XOR X	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
OR X	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
CP X	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
INC X	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
DEC X	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

CPUコントロール

NOP	00
HALT	76
DI	F3
EI	FB
IM 0	ED 46
IM 1	ED 56
IM 2	ED 5E

16ビット算術演算

X	BC	DE	HL	SP	IX	IY
ADD HL, X	09	19	29	39		
ADD IX, X	DD 09	DD 19		DD 39	DD 29	
ADD IY, X	FD 09	FD 19		FD 39		FD 29
ADC HL, X	ED 4A	ED 5A	ED 6A	ED 7A		
SBC HL, X	ED 42	ED 52	ED 62	ED 72		
INC X	03	13	23	33	DD 23	FD 23
DEC X	0B	1B	2B	3B	DD 2B	FD 2B

エクスチェンジ

EX AF, AF'	08
EX DE, HL	EB
EX (SP), HL	E3
EX (SP), IX	DD E3
EX (SP), IY	FD E3
EXX	D9

アキュムレータ操作

DAA	27
CPL	2F
NEG	ED 44
CCF	3F
SCF	37

ローテート・シフト

X	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)
RLC X	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB d 06	FD CB d 06
RRC X	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB d 0E	FD CB d 0E
RL X	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB d 16	FD CB d 16
RR X	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB d 1E	FD CB d 1E
SLA X	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB d 26	FD CB d 26
SRA X	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB d 2E	FD CB d 2E
SRL X	CB 3F	CB 38	CB 39	CB 3A	CB 38	CB 3C	CB 3D	CB 3E	DD CB d 3E	FD CB d 3E
RLD								ED 6F		
RRD								ED 67		

	A
RLCA	07
RRCA	0F
RLA	17
RRA	1F

ジャンプ，コール，リターン

X	UN COND	C	NC	Z	NZ	PE	PO	M	P	
JP X, nn	C3 n n	DA n n	D2 n n	CA n n	C2 n n	EA n n	E2 n n	FA n n	F2 n n	
JR X, e	18 e−2	38 e−2	30 e−2	28 e−2	20 e−2					
JP (HL)	E9									
JP (IX)	DD E9									
JP (IY)	FD E9									
CALL X, nn	CD n n	DC n n	D4 n n	CC n n	C4 n n	EC n n	E4 n n	FC n n	F4 n n	
DJNZ e										10 e−2
RET X	C9	D8	C0	C8	C0	E8	E0	F8	F0	
RETI	ED 4D									
RETN	ED 45									

リスタート

RST 00H	C7
RST 08H	CF
RST 10H	D7
RST 18H	DF
RST 20H	E7
RST 28H	EF
RST 30H	F7
RST 38H	FF

ビット操作

X	A	B	C	D	E	H	L	(HL)	(IX +d)	(IY +d)
BIT 0, X	CB 47	CB 40	CB 41	CB 42	CB 43	CB 44	CB 45	CB 46	DD CB d 46	FD CB d 46
BIT 1, X	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	DD CB d 4E	FD CB d 4E
BIT 2, X	CB 57	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	DD CB d 56	FD CB d 56
BIT 3, X	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	DD CB d 5E	FD CB d 5E
BIT 4, X	CB 67	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	DD CB d 66	FD CB d 66
BIT 5, X	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	DD CB d 6E	FD CB d 6E
BIT 6, X	CB 77	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	DD CB d 76	FD CB d 76
BIT 7, X	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	DD CB d 7E	FD CB d 7E
RES 0, X	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	DD CB d 86	FD CB d 86
RES 1, X	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	DD CB d 8E	FD CB d 8E
RES 2, X	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96	FD CB d 96
RES 3, X	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E	FD CB d 9E

X	A	B	C	D	E	H	L	(HL)	(IX +d)	(IY +d)
RES 4, X	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6	FD CB d A6
RES 5, X	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CA d AE	FD CB d AE
RES 6, X	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6	FD CB d B6
RES 7, X	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE	FD CB d BE
SET 0, X	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6	FD CB d C6
SET 1, X	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE	FD CB d CE
SET 2, X	CB D7	CB D0	CB D1	CB D2	DB D3	CB D4	DB D5	CB D6	DD CB d D6	FD CB d D6
SET 3, X	CB FD	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE	FD CB d DE
SET 4, X	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6	FD CB d E6
SET 5, X	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE	FD CB d EE
SET 6, X	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6	FD CB d F6
SET 7, X	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE	FD CB d FE

入力

IN A, n	DB n
IN A, (C)	ED 78
IN B, (C)	ED 40
IN C, (C)	ED 48
IN D, (C)	ED 50
IN E, (C)	ED 58
IN H, (C)	ED 60
IN L, (C)	ED A2
INI	ED A2
INR	ED B2
IND	ED AA
INDR	ED BA

出力

OUT n, A	D3 n
OUT (C), A	ED 79
OUT (C), B	ED 41
OUT (C), C	ED 49
OUT (C), D	ED 51
OUT (C), E	ED 59
OUT (C), H	ED 61
OUT (C), L	ED 69
OUTI	ED A3
OTIR	ED B3
OUTD	ED AB
OTDR	ED BB

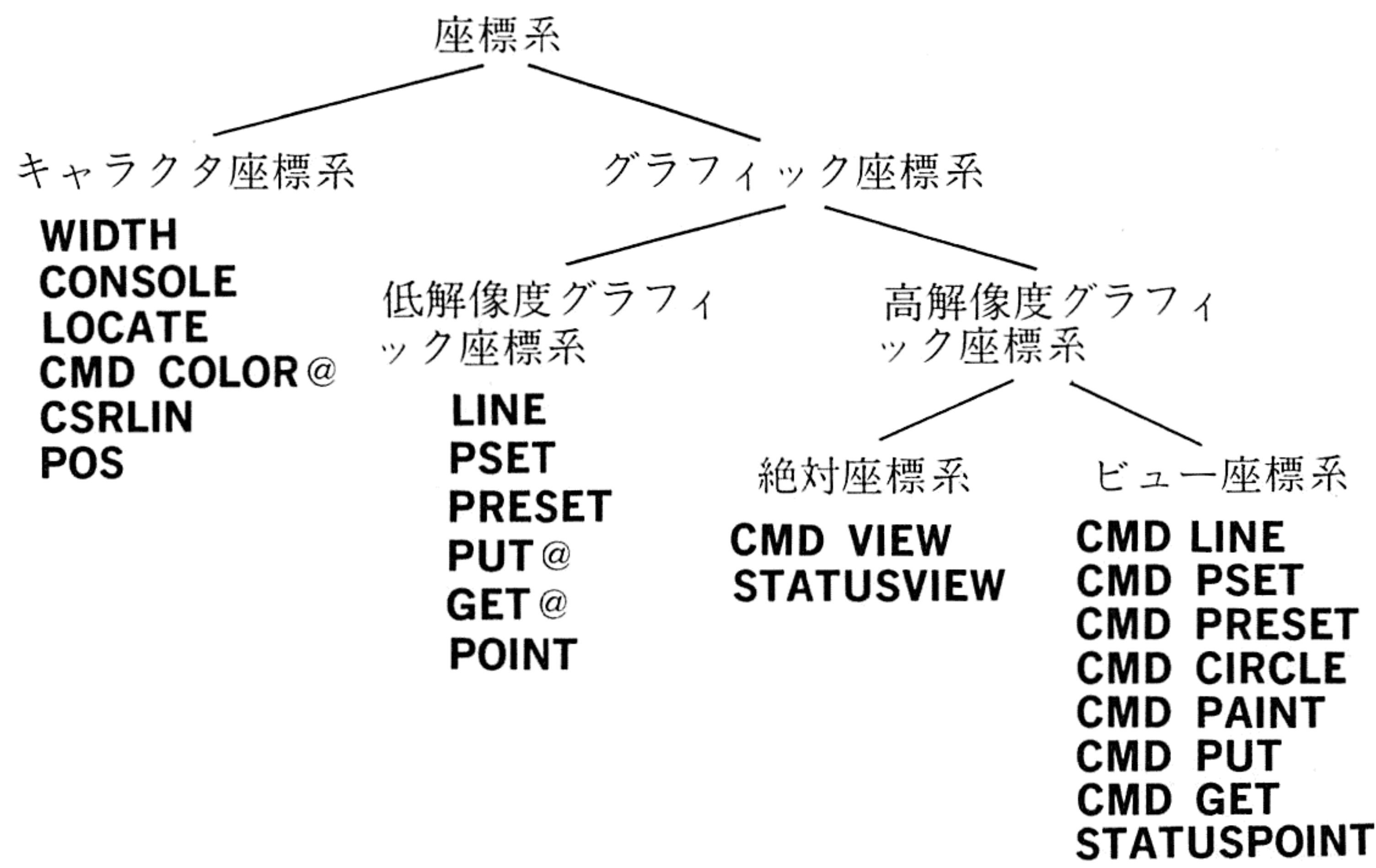
付録12　機械語のサブルーチン

ROM に書き込まれている機械語のサブルーチンのいくつかをあげます。

機　　能	サブルーチンの使い方
画面に 1 文字表示する。	Aレジスタにキャラクタコードを入れてから CALL 257Hを実行する。
キー入力があるまで待って，1 文字入力されたら戻る。	CALL F75Hを実行すると，Aレジスタに入力されたキーの キャラクタコードを入れて戻る。
キー入力があるかどうかを調べる。	CALL F7BHを実行すると，キー入力があれば Aレジスタに入力されたキーの キャラクタコード，キー入力がなければキャリフラグをセットして戻る
プリンタに 1 文字出力する。	Aレジスタにキャラクタコードを入れてから CALL D60Hを実行する。
画面をプリンタに出力する。	CALL 124AHを実行する。
文字列を画面に表示する。	HLレジスタに文字列の先頭アドレスを入れてCALL 52EDHを実行する。文字列の最後には，00Hをつける。
BASICのコマンド入力待ち状態にする。（ウォームスタート）	JP 8Hを実行する。
モニタの コマンド 入力待ち状態にする。	JP 5C66Hを実行する。
画面に〝？〟を 表示してモニタのコマンド入力待ち 状態にする。	JP 5C2CHを実行する。

これらのサブルーチンの使い方は，**第10章モニタ**の **S** コマンドと **G** コマンドの使い方のところで示したサンプルプログラムを参照してください。

付録13　座標系

- 座標系
  - キャラクタ座標系
    - WIDTH
    - CONSOLE
    - LOCATE
    - CMD COLOR@
    - CSRLIN
    - POS
  - グラフィック座標系
    - 低解像度グラフィック座標系
      - LINE
      - PSET
      - PRESET
      - PUT@
      - GET@
      - POINT
    - 高解像度グラフィック座標系
      - 絶対座標系
        - CMD VIEW
        - STATUSVIEW
      - ビュー座標系
        - CMD LINE
        - CMD PSET
        - CMD PRESET
        - CMD CIRCLE
        - CMD PAINT
        - CMD PUT
        - CMD GET
        - STATUSPOINT

付録14　高解像度グラフィックスのモード

CMD SCREEN の起動時の設定値

　　CMD SCREEN 0, 0, 7

CMD COLOR の起動時の設定値

　　CMD COLOR 3, 0

グラフィックモード	モノクロモード	アトリビュートカラーモード	4色カラーモード	
			4色カラーモード0	4色カラーモード1
設定方法	CMD SCREEN 0	CMD SCREEN 1	CMD SCREEN 2	CMD SCREEN 3
解像度	640×200ピクセル		320×200ピクセル	
カラーナンバー (　)内は起動時に設定されている色	0：黒 1：選択色(白)	0：選択色(黒) 1：テキスト画面の色に依存する	0：黒 1：緑 2：赤 3：選択色(青)	0：青 1：シアン 2：マゼンタ 3：選択色(黒)

付録15　ソフトウェア上の一般的注意事項

1．**PC-8001** 用のプログラムを使用するときは，ディップスイッチ8をonにセットして **N-BASIC** モードで使用してください。

PC-8001 用のプログラムを **N80-BASIC** モードで使用した場合，正常に動作しないことがあります。その原因として考えられることは，①機械語のサブルーチンなどが，**N80-BASIC** のワークエリアを使用している，②プログラムのサイズや，変数領域が大きすぎて（大きな配列を使用している等），実行すると Out of memory エラーを生ずる，などがあります。

2．**PC-8001** 用のプログラムのうち割り込みを使っているものは **MKII** を **N-BASIC** モードにしても使えないことがあります。

3．**N80-BASIC, N80DISK-BASIC** において，6000H 番地から 7FFFH 番地に書かれた機械語プログラムを **USR** 関数で呼び，その値を **CMD** のついた命令と **STATUS** の中で使うような場合は，①のようにします。②のように直接使うことはできません。

①正しい例

```
10 DEFUSR=&H6808
15 TMP=USR(0)
20 CMD VIEW(TMP, 1)-(100, 100)
```

②誤っている例

```
10 DEFUSR=&H6808
20 CMD VIEW(USR(0), 1)-(100, 100)
```

4．市販の **PC-8001** 用のプログラムで，ブザーを使っているものの中に，カレンダ時計を狂わせてしまうものがあります。このようなときは，もう一度カレンダ時計の値を設定すれば正しい状態に直ります。

5．**N80 DISK-BASIC** では，ファイルをオープンしたときには，フロッピィディスクを取り出す前に必ずクローズしてください。クローズせずにフロッピィディスクを取り出すとファイルがこわれてしまいま

す。

6. **DISK-BASIC**では，フロッピィディスクを取り出す前に必ず**REMOVE**を実行してください。**REMOVE**せずにフロッピィディスクを取り出すとファイルがこわれてしまいます。

7. カセットテープにプログラムをセーブしたときは，必ずベリファイしてください。ベリファイは，セーブした直後に行います。セーブした後，**RUN**してしまうとその後ベリファイしてもメモリ上のプログラムと，カセットテープ上のプログラムは一致しないことがあります。

8. カセットテープにセーブするプログラムの最後には必ず**END**をつけてください。**END**がないと，そのプログラムをロードしても実行できないことがあります。

9. ターミナルモードを使用するときボーレートを600bps. 以上に設定すると，受信エラーになることがあります。

10. N_{80} DISK-BASICでプログラムの実行中またはロード中に“Out of memory”エラーが出るときは，起動時に“How many files?”に対して答える，「同時にオープンするファイル数」を減らしてください。

周辺装置の接続説明図

PC-8001MKII

スロット1

スロット2

カップラ

カップラケーブル
PC-8895(RS232Cケーブル)
オプション

ミニディスクユニット
ケーブル PC-8898：
(PC-8031-2W,
1W用オプション)
PC-8098：
(PC-80S31用)

拡張ドライブ信号
CN2

FDD制御信号
CN1

ミニディスクユニット

標準ディスクユニット用
インタフェースボード

拡張用ディスク
ユニットケーブル

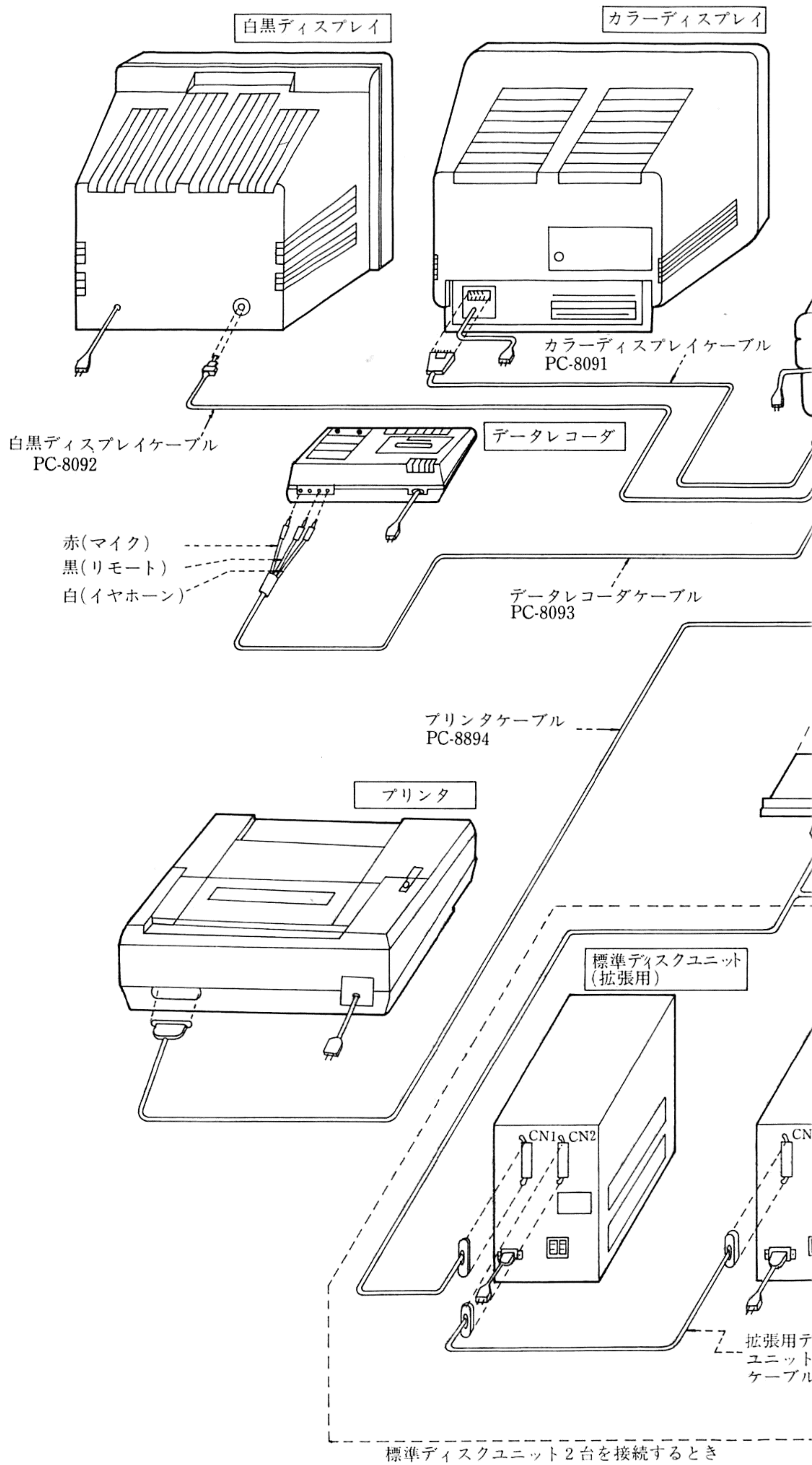
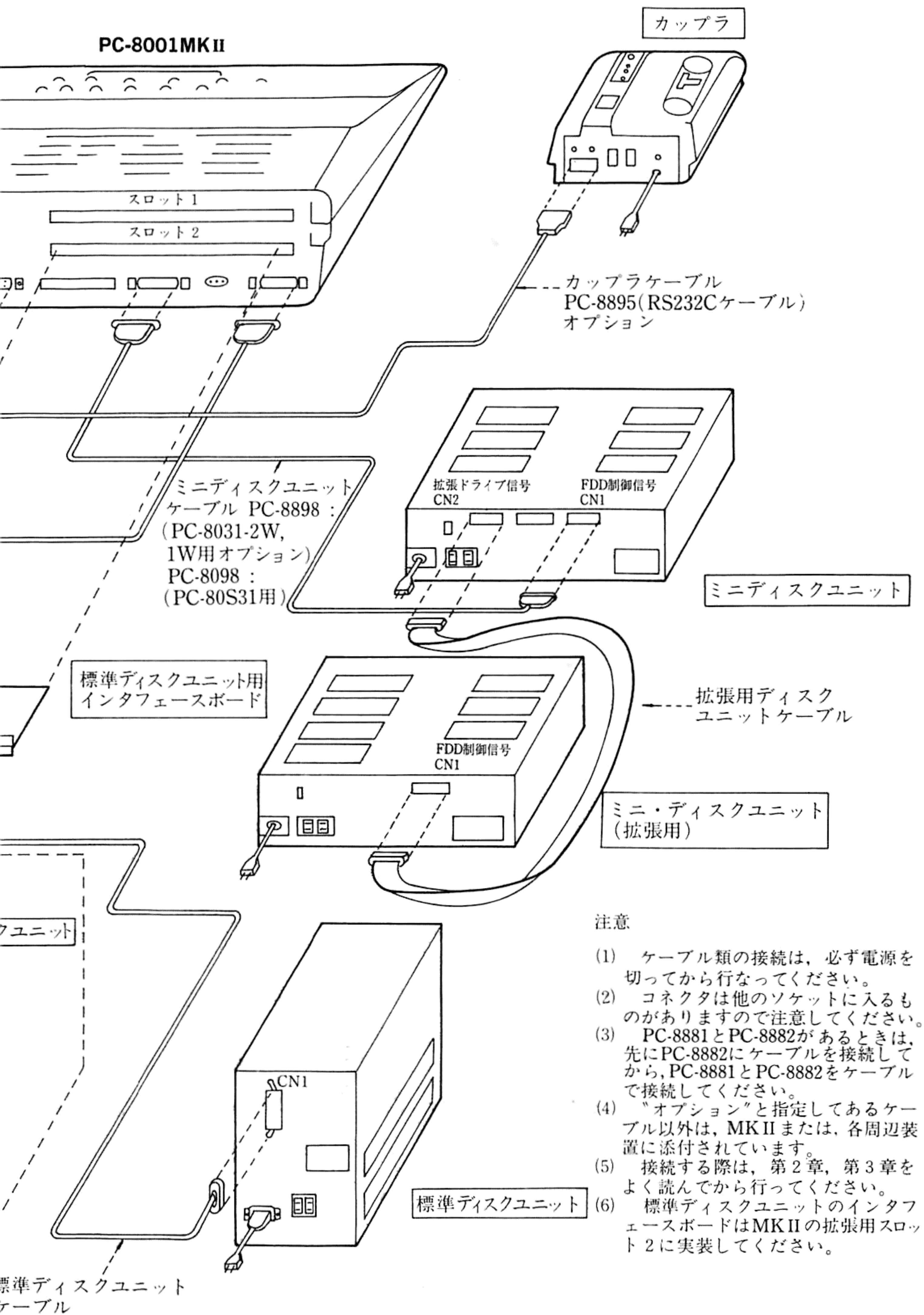
FDD制御信号
CN1

ミニ・ディスクユニット
(拡張用)

クユニット

CN1

標準ディスクユニット

票準ディスクユニット
ケーブル

注意

(1) ケーブル類の接続は，必ず電源を切ってから行なってください。
(2) コネクタは他のソケットに入るものがありますので注意してください。
(3) PC-8881とPC-8882があるときは，先にPC-8882にケーブルを接続してから，PC-8881とPC-8882をケーブルで接続してください。
(4) "オプション"と指定してあるケーブル以外は，MKIIまたは，各周辺装置に添付されています。
(5) 接続する際は，第2章，第3章をよく読んでから行ってください。
(6) 標準ディスクユニットのインタフェースボードはMKIIの拡張用スロット2に実装してください。